Documentation on SLIT-OFFSET
(Combined offset program for use with ALFOSC / FIES).
Version 2.0

Jacob Wang Clasen
Nordic Optical Telescope

February 28, 2000

Contents

Chapter 1

Description

The purpose of this program (SLIT-OFFSET) is to facilitate the task of offsetting the telescope/guideprobe in order to place an object on the slit/aperture of the ALFOSC spectrograph and on the fiber of FIES. The program is designed as a stand-alone program, i.e. it is self-contained and is executed from the shell command line.
It's usage in combination with the CCD aquisition program (BIAS) works via a small modifocation in BIAS. This serves to obtain the objects position on the CCD and then call SLIT-OFFSET with these parameters. It is not possible to perform other actions within BIAS while SLIT-OFFSET is executing. SLIT-OFFSET is called with the following parameters (also see 'usage')

- IP-number
  The IP number of the TCS/VME computer.

- Accesscode
  The current accesscode for communicating with the TCS.

- X position of star
  The X position of the object on the CCD.

- Y position of the star
  The Y position of the object on the CCD.

- Slit number
  The number of the slit currently in the light path in ALFOSC. Numbered from 1 to 8 (both included).

- CCD Number
  The number of the CCD detector used. This is used for pixed scale and orientation purposes.

The program now takes action as follows

- Search the file `slitpos.def` to obtain the CCD X/Y position of the slit in use.

- Makes a connection to the RPC server specified in argument 'IP-number'. If this does not succeed, the program returns the exit status -2.
- Gets the current field rotation from the telescope and checks if the telescope is being autoguided.
- Switches autoguiding off.
- Determines the the amount to move the telescope in seconds of arc in a non rotated coordinate system, i.e. a system with the normal ALFOSC field of -90°. The $zeropoint_{field}$ has therefore the value 90.
- Finds the amount to move the telescope in the coordinate system rotated "field" degrees:

$$\Delta_\alpha = \Delta_x \cdot cos(field + zeropoint_{field}) - \Delta_y \cdot sin(field + zeropoint_{field})$$

$$\Delta_\delta = \Delta_x \cdot sin(field + zeropoint_{field}) + \Delta_y \cdot cos(field + zeropoint_{field})$$

- The value of the δ movement is negated because increasing values of CCD Y corresponds to descreasing declination.
- The telescope is moved.
- The amount to move the guideprobe is determined as

$$X_{new} = X_{old} - \frac{\Delta_\delta}{0.0073 \frac{asec}{step}}$$

$$Y_{new} = Y_{old} - \frac{\Delta_\alpha}{0.0073 \frac{asec}{step}}$$

- The new guideprobe position is checked to be within limits.
- The guideprobe is moved.
- Waits until the guideprobe has reached its new position.
- Waits until the telescope is tracking.
- Switches autoguiding on.
- Returns to caller with exit status 0 (zero).

Chapter 2

Changes since last version

From version 0.1 the following things have been changed

- The check of the guideprobe limits have been moved so that the check is being performed before any telescope or guideprobe movements.
- Error message when both X and Y is outside limits have been added.
- Selection of pixel scale and field offset value based on the CCD number. Now the program can be used for FIES object aquisition with Stancam.

Chapter 3

Compilation

Change directory (command `cd`) to `/home/software/rpc/slitoffset`. The directory should contain the files

- `slit-offset.c`
- `Makefile`
- `slitpos.def`

A number of other files are needed to be linked with SLIT-OFFSET and these should be located in a known directory before proceeding. The files needed are

- hostintf5.o, hostintf5_clnt.o, hostintf5_xdr.o
- gettcs5.h

Edit the makefile to make `RDIR` point to the directory containing above files. Type `make slit-offset`. The directory will now also contain the file

- `slit-offset`

The compilation is completed.

Chapter 4

Installation

Copy the file `slit-offset` to the `/usr/local/bin` directory. The BIAS program expect to find `slit-offset` in this directory.
Edit the file `slitpos.def` to contain the correct slit X/Y positions and copy the file to the directory from where the `startba` command is given. Currently `~obs1/alfosc` or `~obs2/alfosc` and `~obs1/stancam` or `~obs2/stancam` for use with FIES/Stancam.

Chapter 5

Usage

The normal usage of SLIT-OFFSET is hidden internally in BIAS. BIAS makes a system call (system(3)) to call SLIT-OFFSET.
In the following, the procedure for placing an object on the slit is described. This is done from within BIAS.

- Make an image of the object which should be offset to the slit.
- Type `imexa` and place the cursor on the object which should be placed on the slit.
- Press `t` or `y`.
- You will be asked to enter the slit number. Do so.
  Slit number 1 (one) is the slit selected by pressing the top button in the ALFOSC user interface / slit selection.
- Wait for the message `Look at the guidestar monitor, and wait for the guidestar to be centered in the box`. Check the guidestar.
- Press '`q`' to get out of imexamine.
- Make an image thrugh the slit to check that the object is on the slit.

The `t` will return the actual x,y position of the cursor. Use `y` to return the centroid of a Gauss fit of the object close to the cursor position.

Be aware of that prior to this procedure, the apropriate field rotation should be set. A suitable guidestar for this field rotation should be found and the telescope should be autoguiding on this guidestar.
The program can also be executed manually from the commandline, but this is only intended for testing. The full usage of the program is

- `slit-offset` IP-number accesscode star-x-position star-y-position slit-number CCD-number

,where
accesscode is the currently valid TCS accesscode (command `sh 84` on TCS,
star-x-position is the X position of the object on the CCD,
star-y-position is the Y position of the object on the CCD,
slit-number is the slit number in use in ALFOSC.
CCD-number is 1 for Stancam and 7 for ALFOSC.

Example:

```
slit-offset 161.72.5.7 325911657 734 288 1 7
```

The file `slitpos.def` is required for SLIT-OFFSET to work. It is located in the alfosc directory under the observer account (`~obs1/alfosc/slitpos.def` or `~obs2/alfosc/slitpos.def` for use with ALFOSC and `~obs1/stancam/slitpos.def` or `~obs2/stancam/slitpos.def` for use with FIES/Stancam). It contains information on the position of the slits on the CCD and has the following format

```
1 1020 1024
2 1102 993
3 1024 1024
```

One line describes the position of one slit. In this example the position of slit number 2 is $(x, y) = (1102, 993)$.

The program returns an exit status according to the success of the execution. A number of things can go wrong, and the easiest way to report it back to the caller is by using the exit status. The following values will (can) be returned

- 0
  Normal termination (all ok).
- -1
  Unable to establish RPC connection to TCS.
- -2
  Telescope is not being autoguided.
- -3
  Guideprobe movement will result in X and Y value out of range.
- -4
  Guideprobe movement will result in X value out of range.
- -5
  Guideprobe movement will result in Y value out of range.

Chapter 6

Bugs

- The actual (measured TGT X,Y) differs from the desired by between 0 and 10 pixels. Possibly the various scales used have not been determined well enough.

Chapter 7

Program listing.

7.1 Main program.

```
#include <stdio.h>
#include <math.h>
#include <time.h>
#include "../hostintf5/gettcs5.h"
/* Functions used:
   FieldRotationDelta
   MoveRADelta
   MoveDECDelta
   AutoGuideOff
   AutoGuideOn
   MoveGuideprobeDeltaX
   MoveGuideprobeDeltaY
   MoveGuideprobeX;
   MoveGuideprobeY;

   Usage: offset IP-number accesscode star-x-position star-y-position slit-number CCD-number

   - IP-number: IP address of the TCS computer.
   - accesscode: Required accesscode.
   - star-x-position: X position of the star to be positioned.
   - star-y-position: Y position of the star to be positioned.
   - slit-number: Number of the slit used ('1' is the slit in the first line
                  of the ALFOSC slit definition file "slit.def").
   - CCD-number: Number of the CCD used for aquisition (because the scale is
                 different). Can be one of 1 (Stancam), 4 (HiRAC), 7 (ALFOSC)

   Return values:
     0: Normal termination (all ok).
    -1: Unable to establish RPC connection to TCS.
    -2: Telescope is not being autoguided.
    -3: Guideprobe movement will result in X and Y value out of range.
    -4: Guideprobe movement will result in X value out of range.
    -5: Guideprobe movement will result in Y value out of range.
    -6: Timed out.

15/2 2000: Added selection of pixel scale and field offset value based on
           the CCD number.
*/
#define DEBUG TRUE
#define TIMEOUT 20
/*****************************************************************/
void get_current_slit(void)
{
```

```
}
/******************************************************************/
void read_slit_position(int pos, double *x, double *y)
{
  char str[128];
  FILE *f1;
  char *token;
  char *token_id;

  strcpy(str,"slitpos.def");

  if ( (f1 = fopen(str,"r") ) == NULL) {
    fprintf(stderr,"Error opening %s\n",str);
    return;
  }

  while ( fgets(str,128,f1) != NULL ) {
    token = strtok(str," \n");

    if ( atoi(token) == pos ) {
      token_id = token;
      token = strtok(NULL," \n");
      *x = atof(token);

      token = strtok(NULL," \n");
      *y = atof(token);
    }
  }
  fclose(f1);
  return;
}
/******************************************************************/

main(argc, argv)
int argc;
char **argv;
{
  double x1, y1; /* The star selected */
  double SlitX, SlitY;
  int SlitNo, CCDno;
  double Xmovement, Ymovement;
  int OldGuideprobeX, OldGuideprobeY;
  int NewGuideprobeX, NewGuideprobeY;
  int AutoGuiding; /* 1: autoguiding 0: not autoguiding */
  double field; /* Field rotation */
  double field_off = 90.0; /* Field zeropoint offset (field -90 is zero)*/
  double scalefactor = 0.007153; /* asec pr. ag step */
  /*double ccd_scale = 0.1854;  asec pr. pixel */
  double ccd_xscale = 0.18595; /* asec pr. pixel in x */
  double ccd_yscale = 0.18684; /* asec pr. pixel in y*/
  int TCS_accesscode, T;
  int *FieldRotDeltaRes,
   *RADeltaRes,
   *DECDeltaRes,
   *AutoGuideOffRes,
   *AutoGuideOnRes,
   *GuideprobeDeltaXRes,
   *GuideprobeDeltaYRes,
   *GuideprobeXRes,
   *GuideprobeYRes;
  char *getserver;

  fieldrotdeltatype a, *fieldrotdelta_info = &a;
  radeltatype b, *radelta_info = &b;
  decdeltatype c, *decdelta_info = &c;
  agontype d, *agon_info = &d;
  agofftype e, *agoff_info = &e;
  guideprobedeltaxtype f, *guideprobedeltax_info = &f;
```

```
guideprobedeltaytype  g, *guideprobedeltay_info = &g;
guideprobextype h, *guideprobex_info = &h;
guideprobeytype i, *guideprobey_info = &i;
allstatustype *telescope;

getserver      = argv[1];
TCS_accesscode = atoi(argv[2]);
x1             = atof(argv[3]);
y1             = atof(argv[4]);
SlitNo         = atoi(argv[5]);
CCDno          = atoi(argv[6]);

/* Set the CCD pixel scale depending on the CCDno */
if ( CCDno == 1 ) {
  if (DEBUG) printf("CCD is no. %d\n",CCDno);
  ccd_xscale = ccd_yscale = 0.176;
  field_off = 0.0; /* Nominal field is 0 -> 0+0=0 */
} else
if ( CCDno == 4 ) {
  if (DEBUG) printf("CCD is no. %d\n",CCDno);
  ccd_xscale = ccd_yscale = 0.110;
  field_off = 0.0; /* Nominal field is 0 -> 0+0=0 */
} else
if ( CCDno == 7 ) {
  if (DEBUG) printf("CCD is no. %d\n",CCDno);
  ccd_xscale = 0.18595;
  ccd_yscale = 0.18684;
  field_off = 90.0; /* Nominal field is -90 -> -90+90=0 */
}

fieldrotdelta_info->AccessCode = TCS_accesscode;
radelta_info->AccessCode = TCS_accesscode;
decdelta_info->AccessCode = TCS_accesscode;
agon_info->AccessCode = TCS_accesscode;
agoff_info->AccessCode = TCS_accesscode;
guideprobedeltax_info->AccessCode = TCS_accesscode;
guideprobedeltay_info->AccessCode = TCS_accesscode;
guideprobex_info->AccessCode = TCS_accesscode;
guideprobey_info->AccessCode = TCS_accesscode;

read_slit_position(SlitNo, &SlitX, &SlitY);

if (InitRPCService(getserver) == NULL) {
 fprintf(stderr,"Unable to read TCS!\n");
 exit(-1);
}

if ((telescope = GetTelescope()) == NULL) exit(1);
AutoGuiding = telescope->AutoGuiding;
OldGuideprobeX = telescope->ActualGuideProbeX;
OldGuideprobeY = telescope->ActualGuideProbeY;
field = telescope->ActualFieldRotation;
  fprintf(stdout,"Field rotation: %4.2f\n",field);
if (!AutoGuiding) {
 fprintf(stderr,"You have to be autoguiding...\n");
 exit(-2);
}

/* SIGN! Is correct */
/* The "-" before field makes it right when counter clockwise rotation */
/*  is assumed positive. If opposite, remove the "-". */
Ymovement = ((SlitY - y1) * ccd_yscale);

if ( CCDno == 1 ) {
  Xmovement = -((SlitX - x1) * ccd_xscale);
} else
if ( CCDno == 7 ) {
  Xmovement = ((SlitX - x1) * ccd_xscale);
```

```
  }

  /* This is the amount to move the guideprobe opposite the tel. */
  /* This is only true if the angle between x,y and alpha,delta is 0 /*
  /* In any other case, calculate what the angle does */
  /* SIGN! */

  /* Sign is correct */
  if ( CCDno == 1 ) {
    NewGuideprobeX = OldGuideprobeX + (int)(Xmovement / scalefactor);
    NewGuideprobeY = OldGuideprobeY - (int)(Ymovement / scalefactor);
  } else
  if ( CCDno == 7 ) {
    NewGuideprobeX = OldGuideprobeX - (int)(Ymovement / scalefactor);
    NewGuideprobeY = OldGuideprobeY - (int)(Xmovement / scalefactor);
  }

  /* Check if within limits */
  if ( ((NewGuideprobeX <= 1100.0) || (NewGuideprobeX >= 225000.0)) &&
       ((NewGuideprobeY <= 1100.0) || (NewGuideprobeY >= 165000.0)) ) {
fprintf(stdout,"Guideprobe X and Y outside limits...press 'q'.\n");
exit(-3);
  }
  if ( (NewGuideprobeX <= 1100.0) || (NewGuideprobeX >= 225000.0) ) {
   fprintf(stderr,"Guideprobe X outside limits...press 'q'.\n");
   exit(-4);
  }
  if ( (NewGuideprobeY <= 1100.0) || (NewGuideprobeY >= 165000.0) ) {
   fprintf(stderr,"Guideprobe Y outside limits...press 'q'.\n");
   exit(-5);
  }

  /* Swich off autoguiding */
  fprintf(stdout,"Switching off autoguiding...\n");
  if ((AutoGuideOffRes = AutoGuideOff(agoff_info)) == NULL) exit(1);

  radelta_info->RAdelta = Xmovement * cos((field + field_off)*(M_PI/180.0)) -
                          Ymovement * sin((field + field_off)*(M_PI/180.0));
  decdelta_info->DECdelta = Xmovement * sin((field + field_off)*(M_PI/180.0)) +
                            Ymovement * cos((field + field_off)*(M_PI/180.0));
  /* We need to invert the delta motion because south (DEC--) is y-value++ */
  decdelta_info->DECdelta = -(decdelta_info->DECdelta);

  /* Move the telescope */
  if (DEBUG) {
    fprintf(stdout,"Field: %4.2f   Slitpos: x:%4.2f y:%4.2f\n",field,SlitX,SlitY);
    fprintf(stdout,"Obj pos: x:%4.2f y:%4.2f\n",x1, y1);
    fprintf(stdout,"Telescope movement in CCD sys.: alpha: %4.2f   delta: %4.2f\n",
     Xmovement, Ymovement);
  }
  if (DEBUG)
    fprintf(stdout,"Telescope movement: alpha: %4.2f   delta: %4.2f\n",
radelta_info->RAdelta,decdelta_info->DECdelta);

  fprintf(stdout,"Moving the telescope...\n");
  if ((RADeltaRes = MoveRADelta(radelta_info)) == NULL) exit(1);
  if ((DECDeltaRes = MoveDECDelta(decdelta_info)) == NULL) exit(1);

  if (DEBUG) {
    fprintf(stdout,"Old XY position:\n");
    fprintf(stdout,"           X: %d   Y: %d\n",OldGuideprobeX,OldGuideprobeY);
    fprintf(stdout,"New position in CCD system:\n");
    fprintf(stdout,"           X: %d Y: %d\n",
          NewGuideprobeX, NewGuideprobeY);
  }

  guideprobex_info->DesiredX = NewGuideprobeX;
  guideprobey_info->DesiredY = NewGuideprobeY;
```

```
  /* Make the offset */
  fprintf(stdout,"Moving the guideprobe...\n");
  if ((GuideprobeXRes = MoveGuideprobeX(guideprobex_info)) == NULL) exit(1);
  if ((GuideprobeYRes = MoveGuideprobeY(guideprobey_info)) == NULL) exit(1);

  T = time(NULL); /* Set T = the time now */

  do {
    if ((telescope = GetTelescope()) == NULL) exit(1);
    usleep(500000);
    if (time(NULL) - T > TIMEOUT) exit(-6); /* Are we timed out? */
    if (DEBUG) {
      fprintf(stdout,"Waiting for guideprobe...\n");
      fprintf(stdout,"X: %d  Y: %d\n",telescope->ActualGuideProbeX,telescope->ActualGuideProbeY);
    }
  } while ( (telescope->ActualGuideProbeX != NewGuideprobeX) &&
          (telescope->ActualGuideProbeY != NewGuideprobeY) );

  T = time(NULL);

  do {
    if ((telescope = GetTelescope()) == NULL) exit(1);
    usleep(500000);
    if (time(NULL) - T > TIMEOUT) exit(-6); /* Are we timed out? */
    if (DEBUG) {
      fprintf(stdout,"Waiting for telescope...\n");
    }
  } while (telescope->TelescopeModeNumber != 4);

  /* Swich autoguiding back on */
  fprintf(stdout,"Turning autoguiding back on...\n");
  if ((AutoGuideOnRes = AutoGuideOn(agon_info)) == NULL) exit(1);

  fprintf(stdout,"Look at the guidestar monitor and wait for\n");
  fprintf(stdout," the guidestar to be centered in the box, then press 'q'\n");

  exit(0);
}
```

7.2 Makefile.

```
SHELL=/bin/sh
#DF = _CLASSIC_ID_TYPES
DF = __STDC__
CC = gcc
LD = ld
ODIR = .
RDIR = ../rpc/hostintf5
PDIR = .
#TMPD = /dd/TEMP
CFLAGS = -O -c -D$(DF) #-Wall
LDFLAGS = -lm

RPCLIB   = rpcsvc

LINKLIBS = -l$(RPCLIB)

ROTATION  = rotation
OFFSET = offset

SERVEROBJS = $(RDIR)/hostintf5_svc.o $(RDIR)/hostintf5_xdr.o
CLIENTOBJS = $(RDIR)/hostintf5.o $(RDIR)/hostintf5_clnt.o $(RDIR)/hostintf5_xdr.o

HDRS = $(RDIR)/hostintf5.h $(RDIR)/gettcs5.h
```

```
ROTATIONOBJS = $(CLIENTOBJS) $(PDIR)/rotation.o

OFFSETOBJS = $(CLIENTOBJS) $(PDIR)/offset.o

$(ROTATION): $(ROTATIONOBJS)
$(CC) $(LDFLAGS) -o $@ $(ROTATIONOBJS)

$(OFFSET): $(OFFSETOBJS)
$(CC) $(LDFLAGS) -o $@ $(OFFSETOBJS)

$(CLIENTOBJS): $(HDRS)
```

7.3 Code inserted into BIAS.

File `imex.c` line 139 (in function `void imex(void)` in the big switch statement that dispaches the different keystrokes used within imexamine).

```
case 't' : {offset_to_slit();
                        sleep(4);break;}
case 'y' : {offset_to_slit_gauss();
          sleep(4);break;}
```

In file `imex.c` line 364 the following new function is inserted. It is also inserted into the file `bias.h` as a prototype.

```
/***************************************************************************/
void offset_to_slit(void)
{
char *str[60];
unsigned short int abuf[10202];
int slit_no = 1;
int returnval, tmp = 1; /* tmp: Default slit number */

if (cursor_set)
erase_cursor(last_cur_x, last_cur_y, last_cur_size, last_cur_type);
set_cursor((double)sao_xpos, (double)sao_ypos, aperture, RECTANGLE);
last_cur_x = sao_xpos;
last_cur_y = sao_ypos;
last_cur_size = aperture;
last_cur_type = RECTANGLE;
cursor_set = 0x7f;

getvalues(sao_xpos-aperture/2, sao_ypos-aperture/2, \
          sao_xpos+aperture/2, sao_ypos+aperture/2, abuf);

jprintf("\n\nSlitoffset: Starposition: X: %d  Y: %d\n",(sao_xpos*xbin)+(xoff-1),
                                      (sao_ypos*ybin)+(yoff-1)  );

chg_int(&tmp,"\nSlit in position: ");
slit_no = tmp;

stop_timer_real();

sprintf(str,"/usr/local/bin/slit-offset 161.72.5.7 %d %d %d %d 7", \
             TCS_accesscode, (sao_xpos * xbin)+(xoff-1), \
             (sao_ypos * ybin)+(yoff-1), slit_no);
returnval = system(str);

/*
```

```
switch (returnval) {
case -1: fprintf(stderr,"Unable to establish RPC connection to TCS.\n");
case -2: fprintf(stderr,"Telescope is not autoguiding, please do so.\n");
case -3: fprintf(stderr,"Guideprobe is out of X and Y range.\n");
case -4: fprintf(stdout,"Guideprobe is out of X range.\n");
case -5: fprintf(stderr,"Guideprobe is out of Y range.\n");
case -6: fprintf(stderr,"Timed out in talking to TCS.\n");
}
*/
start_timer_real(INTERRUPT);
}
/************************************************************************/

void offset_to_slit_gauss(void)
{
char *str[60];
unsigned short int abuf[10202];
int slit_no = 1;
int returnval, tmp = 1; /* tmp: Default slit number */
double  Xposition, Yposition;

if (cursor_set)
erase_cursor(last_cur_x, last_cur_y, last_cur_size, last_cur_type);
set_cursor((double)sao_xpos, (double)sao_ypos, aperture, RECTANGLE);
last_cur_x = sao_xpos;
last_cur_y = sao_ypos;
last_cur_size = aperture;
last_cur_type = RECTANGLE;
cursor_set = 0x7f;

getvalues(sao_xpos-aperture/2, sao_ypos-aperture/2, \
          sao_xpos+aperture/2, sao_ypos+aperture/2, abuf);

get_center_at(&Xposition, &Yposition, -1, -1);

jprintf("\n\nSlitoffset: Starposition: X: %4.2f Y: %4.2f\n",\
                   (Xposition*xbin)+(xoff-1), (Yposition*ybin)+(yoff-1)  );

chg_int(&tmp,"\nSlit in position: ");
slit_no = tmp;

stop_timer_real();

sprintf(str,"/usr/local/bin/slit-offset 161.72.5.7 %d %4.2f %4.2f %d 7", \
            TCS_accesscode, (Xposition * xbin)+(xoff-1), \
            (Yposition * ybin)+(yoff-1), slit_no);
returnval = system(str);

start_timer_real(INTERRUPT);
}
/************************************************************************/
```