LIMITS TO OBSERVING TIMES FOR ALTAZIMUTH TELESCOPES

R.A. Laing
Royal Greenwich Observatory,
Madingley Road,
Cambridge CB3 0EZ.

Technical Note 71

Abstract

The length of time for which an object can be observed with Altazimuth telescopes such as the WHT is governed by the positions of the limits in Azimuth and Elevation and by the size of the "blind spot" at the zenith. This note gives the formulae by which the effective rising and setting times can be calculated together with specific results for the WHT. In addition, expressions are given for the times at which an instrument rotator on an Altazimuth telescope tracks into its limits.

1 Introduction

The software position limits of the WHT (set slightly within the corresponding hardware pre-limits) allow tracking between Elevations of $10°$ and $90°$ and between Azimuths of $-175°$ and $+355°$ (*i.e.* through a range of $530°$). In addition, an object cannot be followed within $0.21°$ of the zenith because the required rotation rate in Azimuth exceeds the limit of 1 deg s^{-1}. In this note, the time for which an object is visible is calculated as a function of Declination. Sections 2, 3 and 4 describe the effects of the Elevation and Azimuth position limits and the Azimuth speed limit, respectively. The field rotation of an Altazimuth telescope is compensated by rotation of the beam from the telescope or of the instrument itself. In the latter case, restrictions are usually imposed by the need to attach cables to the instrument. For the WHT, the Cassegrain turntable may move between position angles of $-247°$ and $256°$. It is possible for an instrument rotator to hit a limit during normal tracking, and expressions for the sidereal time at which it does so are given in Section 5. The results are summarised in Section 6 and the Appendices give listings of FORTRAN subroutines which may be used to calculate the rising and setting times and turntable limits for a given object.

The WHT is conventionally said to be on the *negative wrap* if its Azimuth is in the range $-175° < A < 0°$ and on the *positive wrap* for $0° \leq A < +355°$. The following notation is used throughout, limits for the WHT being given in brackets. + and − subscripts denote positive and negative limits, respectively.

- ϕ latitude of the telescope ($\phi = 28°45'\,38.1''$)
- h hour angle
- δ Declination
- E elevation ($E_- = 10°$)
- z zenith distance ($z_- = \pi/2 - E_+ = 80°$)
- A Azimuth ($A_+ = 355°$ and $A_- = -175°$)

- ψ parallactic angle
- θ Sky position angle
- ρ mount position angle ($\rho_- = -247°$ and $\rho_+ = 256°$)
- $s = \sin\delta$
- $a = \tan A$
- $c = \cos h$
- $p = \tan\psi$

2 Elevation

It is clear that an object is always above the horizon limit if $\delta > \pi - \phi - z$ and always below it if $\delta < \phi - z$. These Declinations are $+71.24°$ and $-51.24°$, respectively for the WHT. An object with Declinations between these values rises above the elevation limit at an hour angle $-h$, where:

$$\cos z = \cos\delta\cos h\cos\phi + \sin\delta\sin\phi$$

or

$$h = \arccos\left(\frac{\cos z - \sin\delta\sin\phi}{\cos\delta\cos\phi}\right)$$

and sets at hour angle $+h$.

3 Azimuth

Azimuth is given in terms of hour angle, Declination and zenith distance by the pair of expressions:

$$\begin{aligned}\sin A &= -\cos\delta\sin h/\sin z\\ \cos A &= (\sin\delta\cos\phi - \cos\delta\cos h\sin\phi)/\sin z\end{aligned}$$

so

$$a = \tan A = \frac{-\sin h\cos\delta}{\sin\delta\cos\phi - \cos\delta\cos h\sin\phi}$$

Squaring this equation gives a quadratic in $c = \cos h$:

$$Bc^2 + Cc + D = 0$$

where

$$\begin{aligned}B &= \cos^2\delta(1 + a^2\sin^2\phi)\\ C &= -2a^2\sin\delta\cos\delta\sin\phi\cos\phi\\ D &= a^2\cos^2\phi\sin^2\delta - \cos^2\delta\end{aligned}$$

The discriminant of this equation is:

$$C^2 - 4BD = 4\cos^2\delta[a^2(\cos^2\delta - \cos^2\phi) + \cos^2\delta]$$

An object *never* encounters an Azimuth limit if

$$\cos\delta < |\sin A\cos\phi|$$

or

$$\delta > \arccos(|\sin A \cos\phi|)$$

for a telescope in the Northern hemisphere. The WHT's positive Azimuth limit is set at A = 355°, which corresponds to a Declination of $\delta = 85.62°$. For Declinations less than this value, the quadratic has the solutions:

$$h = \pm\arccos\left\{\frac{a^2\sin\delta\sin\phi\cos\phi \pm [a^2(\cos^2\delta - \cos^2\phi) + \cos^2\delta]^{1/2})}{\cos\delta(1 + a^2\sin^2\phi)}\right\}$$

The first $\pm$ in the solution results from the fact that we have used an equation for $\cos h$ and cannot distinguish between positive and negative hour angles. The ambiguity can be sorted out by inspection or more formally by noting that, since $\sin A = -\cos\delta\sin h/\sin z$ with $-\pi/2 \leq \delta \leq \pi/2$ and $0 \leq z \leq \pi/2$, $\sin A$ and $\sin h$ must have opposite signs. The second $\pm$ occurs because an object may pass through the same Azimuth twice. The telescope will, however, be moving in opposite directions in Azimuth in the two cases and a limit forbids motion only in one of these.

In the case of the WHT, the positive and negative limits are at A = 355° and A = −175°, respectively. The Azimuth velocity is given by:

$$A' = \sin\phi - \frac{\cos\phi\cos z\cos A}{\sin z}$$

The Azimuth of an object at A = −175° therefore always *increases* with time and the negative limit is therefore irrelevant for normal tracking. The positive limit can only be encountered when tracking a Northern object below the Pole and this corresponds to the solution:

$$h = +\arccos\left\{\frac{a^2\sin\delta\sin\phi\cos\phi - [a^2(\cos^2\delta - \cos^2\phi) + \cos^2\delta]^{1/2}}{\cos\delta(1 + a^2\sin^2\phi)}\right\}$$

Objects below a certain Declination will encounter the horizon limit first. We equate the expressions for $\cos h$ derived for the Elevation and Azimuth limits. The resulting equation reduces to a quadratic in $s = \sin\delta$:

$$Es^2 + Fs + G = 0$$

where

$$\begin{aligned} E &= 1 + a^2 \\ F &= -2\cos z\sin\phi(1 + a^2) \\ G &= \cos^2 z(1 + a^2\sin^2\phi) - \cos^2\phi \end{aligned}$$

For the WHT, A = 355°, z = 80° and $\phi = 28.76°$. This gives the solution $\delta = 70.66°$. Hence the Azimuth limit is only relevant for Declinations between 70.66° and 85.62° when tracking below the Pole.

4 The zenith blind spot

The condition for the tracking speed in Azimuth to exceed V is $\mu|dA/dh| > V$, where μ is the tracking rate in hour angle (15 arcsec s^{-1}). We define $v = \pm V/\mu$ ($= \pm 240$ for the WHT, the sign being negative to the North of the zenith and positive to the South) and solve the equation $dA/dh = v$ for $\cos h$ to give the hour angle at which an object enters the blind spot.

$$\begin{aligned} v &= dA/dh \\ &= \cos\psi\cos\delta/\sin z \\ &= \frac{\cos\delta(\cos\delta\sin\phi - \sin\delta\cos\phi\cos h}{1 - (\cos\delta\cos h\cos\phi + \sin\delta\sin\phi)^2} \end{aligned}$$

This can be rewritten in the form:

$$Hc^2 + Ic + J = 0$$

where (as before) $c = \cos h$ and

$$\begin{aligned} H &= +v\cos^2\delta\cos^2\phi \\ I &= \cos\delta\sin\delta\cos\phi\,(2v\sin\phi - 1) \\ J &= \cos^2\delta\sin\phi - v + v\sin^2\delta\sin^2\phi \end{aligned}$$

The relevant solutions are:

$$\cos h = \frac{\sin\delta(1 - 2|v|\sin\phi) \pm (\sin^2\delta - 4|v|\sin\phi + 4v^2)^{1/2}}{\pm 2|v|\cos\delta\cos\phi}$$

where the + and − signs refer to objects passing South and North of the zenith, respectively.

Solution of this equation confirms that a simple approximation is sufficient to give the range of Declinations which is affected by the zenith blind spot. Close to transit:

$$dA/dh = \cos\phi/\sin(\delta - \phi)$$

so the condition $|dA/dh| > |v|$ gives $|\delta - \phi| < \cos\phi/v$. For the WHT, this corresponds to $\delta = \phi \pm 0.21^\circ$ or $28.55^\circ < \delta < 28.97^\circ$. In the worst case, the Azimuth velocity first exceeds 1 deg s^{-1} at an hour angle of –31 s, so for all practical purposes it can be thought of as entering the blind spot exactly at transit. The telescope will then rotate as rapidly as possible in order to acquire the object at the other side of the blind spot. This takes roughly 3 minutes at full speed.

5 Rotation limits

The sky position angle, θ, specified for an observation is measured anticlockwise from North. Its definition is instrument-dependent, and the position angle of a spectrograph slit or one of the detector axes is generally used as a reference. The mount position angle, ρ, measures the rotation around the optical axis of the telescope with respect to an arbitrary zero-point. In order to maintain a fixed relation between mount and sky position angle for different configurations, the WHT control system has an offset parameter, θ_0, which is an instrument-dependent constant. The relation between sky PA, mount PA and parallactic angle, ψ is then:

$$\rho = (\theta - \theta_0) - \psi$$

The parallactic angle is given in terms of hour angle and Declination by:

$$\sin\psi = \cos\phi\sin h/\sin z$$

$$\cos\psi = (\cos\delta\sin\phi - \sin\delta\cos\phi\cos h)/\sin z$$

and we will take it to be ranged in $[-\pi, \pi]$. Its variation with hour angle for various Declinations is shown in Figure 1. The parallactic angle is undefined at the zenith ($\delta = \phi$, $h = 0$) and the nadir ($\delta = -\phi$, $h = \pi$). In the special cases $\psi = 0$ and $\psi = \pi$, the hour angle is given by:

$\psi = 0 \Rightarrow h = 0$ for $\delta < \phi$ and $h = \pi$ for $\delta > -\phi$.

$\psi = \pi \Rightarrow h = 0$ for $\delta > \phi$ and $h = \pi$ for $\delta < -\phi$.

Curves of $\psi(h)$ (Figure 1 pass through the following points:

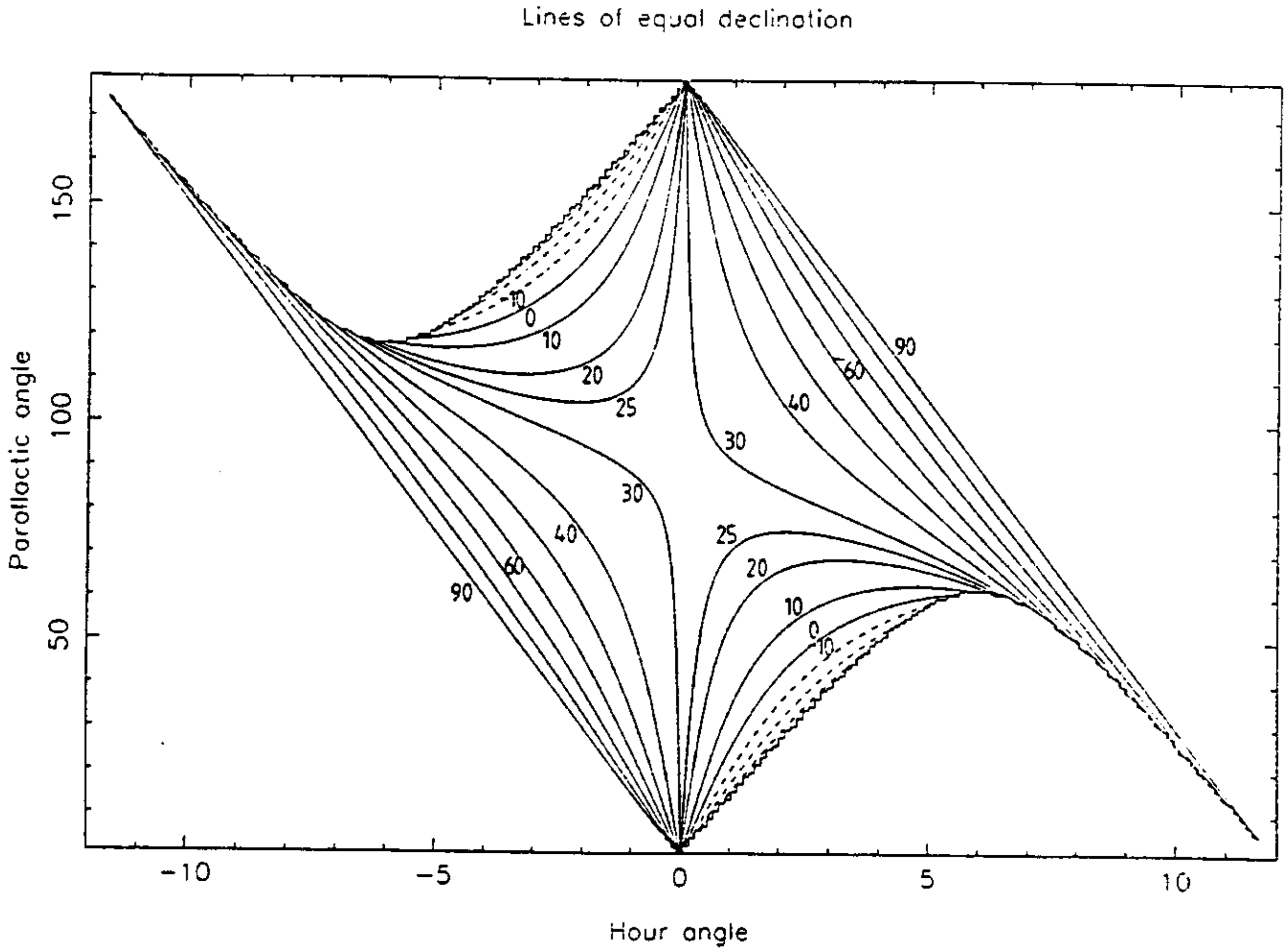

Figure 1: The variation of parallactic angle with hour angle for the latitude of the La Palma Observatory.

$h = \pi$, $\psi = \pi$ and $h = 0$, $\psi = 0$ for $\delta < -\phi$.

$h = \pi$, $\psi = 0$ and $h = 0$, $\psi = 0$ for $-\phi < \delta < \phi$.

$h = 0$, $\psi = 0$ and $h = \pi$, $\psi = 0$ for $\delta > \phi$.

During a sidereal track, a given parallactic angle may occur once, twice or not at all. The condition for 0 or 2 solutions is that the function $\psi(h)$ has extrema. Its derivative at a maximum or minimum is:

$$\psi'(h) = -\cos\phi\cos A/\sin z = 0$$

giving the solutions $\cos A = 0$ or A $= \pm\pi/2$. The expression for $\cos A$ given in Section 3 can be used to reduce this to:

$$\cos h = \tan\delta/\tan\phi$$

which only has solutions if $-\phi \leq \delta \leq \phi$. The maximum occurs for $h > 0$ (A $= \pi/2$), so

$$\sin\psi = -\cos\phi\sin A/\cos\delta = \cos\phi/\cos\delta$$

and the minimum at $h < 0$ (A $= \pi/2$), so

$$\sin\psi = -\cos\phi/\cos\delta$$

Hence:

- There are *no* solutions if $-\phi \leq \delta \leq \phi$ and $\mid \psi \mid < \pi/2$, $\sin\psi > \cos\phi/\cos\delta$ or $\sin\psi < -\cos\phi/\cos\delta$.
- There are *two* solutions if $-\phi \leq \delta \leq \phi$ and $-\cos\phi/\cos\delta < \sin\psi < \cos\phi/\cos\delta$.
- There is *one* solution if $\delta > \phi$ or $\delta < -\phi$.

In order to determine the hour angle corresponding to a given parallactic angle and Declination, we start from the relations for $\sin\psi$ and $\cos\psi$ given earlier. These give:

$$p = \tan\psi = \frac{\cos\phi\sin h}{(\cos\delta\sin\phi - \sin\delta\cos\phi\cos h)}$$

This relation can again be squared and rearranged to give a quadratic in c $= \cos h$:

$$Kc^2 + Lc + M = 0$$

with

$$K = (1 + p^2\sin^2\delta)\cos^2\phi$$
$$L = -2p^2\sin\delta\cos\delta\sin\phi\cos\phi$$
$$M = p^2\cos^2\delta\sin^2\phi - \cos^2\phi$$

There are no solutions if $L^2 - 4KM < 0$, which reduces to:

$$\sin^2\psi > \frac{\cos^2\phi}{\cos^2\delta}$$

as before. Otherwise, the possible solutions are:

$$h = \pm\arccos\left\{\frac{p^2\sin\delta\cos\delta\sin\phi \pm [p^2(\cos^2\phi - \cos^2\delta) + \cos^2\phi]^{1/2}}{\cos\phi(1 + p^2\sin^2\delta)}\right\}$$

As with Azimuth, the use of an equation for $\cos h$ with terms in $\tan^2 \psi$ introduces ambiguities (ψ and $\pm\psi + n\pi$ give the same value of p for any integer n). The sign of the arccos function is determined by noting that $h < 0$ if $\psi < 0$ and $h > 0$ if $\psi > 0$. We therefore take $+\arccos$ for $\psi > 0$ and $-\arccos$ for $\psi < 0$. This leaves the positive and negative roots of the quadratic. For $\delta > \phi$, it is obvious from Figure 1 that there is only one solution for a given parallactic angle. In fact, for $0 \leq \psi \leq \pi$ the roots correspond to parallactic angles ψ and $\pi - \psi$. Similarly for $\delta < -\phi$, $-\pi \leq \psi \leq 0$ and the roots correspond to ψ and $-\psi - \pi$. The situation for $-\phi \leq \delta \leq \phi$ is different: either there are no roots (see above) or both roots correspond to the *same* parallactic angle.

If $\psi = \pm\pi/2$, $\mathrm{p}^2 \rightarrow \infty$ and the roots merge. In this case,

$$h = \pm \arccos(\pm \tan\phi / \tan\delta)$$

with the special cases $h = \pm\pi/2$ at the Poles.

Finally, we consider the sense of rotation.

$$\psi'(h) = -\frac{\cos\phi \cos A}{\sin z} = -\frac{\cos\phi \sin\phi \cos\delta}{\sin^2 z} \left[\frac{\tan\delta}{\tan\phi} - \cos h\right]$$

It is clear from this equation that $\psi'(h)$ is always < 0 for $\delta > +\phi$ and > 0 for $\delta < -\phi$. For intermediate Declinations, the quadratic has two valid solutions. The condition that its discriminant is zero is $\cos h = \tan\delta / \tan\phi$. Thus the positive root has $\cos h > \tan\delta / \tan\phi$, so $\psi' > 0$. Conversely, the negative branch has $\cos h < \tan\delta / \tan\phi$ and $\psi' < 0$.

The limit conditions are as follows (note that the mount PA and parallactic angle increase in opposite senses):

$\delta > \phi$ $\psi' < 0$ for all hour angles, so the rotator can only track into the positive mount PA limit. Since the rotation is monotonic, the limit will eventually be reached from any starting position provided that the telescope is not stopped by the Azimuth or horizon limit, each complete rotation taking 24 hours. It is possible for the telescope to track continuously if $\delta > 85.62°$, in which case the length of the observation is restricted only by the positive mount PA limit (admittedly a somewhat academic example). The positive root of the quadratic applies if $|\psi| > \pi/2$, otherwise the negative root.

$-\phi < \delta < \phi$ There are no solutions if $|\sin\psi| > \cos\phi / \cos\delta$ or $|\psi| > \pi/2$, in which case no limits can be hit; otherwise both roots give valid solutions and either limit can be encountered provided that the initial value of the mount PA is sufficiently close. The condition is:

$$(\theta - \theta_0) - \arcsin(\cos\phi / \cos\delta) < \rho < (\theta - \theta_0) + \arcsin(\cos\phi / \cos\delta)$$

where ρ is the current mount position angle and $(\theta - \theta_0)$ is ranged by adding or subtracting multiples of 2π so that $(\theta - \theta_0) - \rho_\pm$ is in the range $[-\pi, \pi]$. The positive root corresponds to the negative mount position angle limit, and conversely.

$\delta < -\phi$ The positive root is allowed, so $\psi' > 0$ and the rotator can track into the negative mount PA limit. As for $\delta > \phi$, the rotation is monotonic, so the turntable will eventually hit the limit provided that the object does not set first. For a telescope in the Northern hemisphere, such as the WHT, objects with $\delta < \phi$ are not circumpolar, so it is impossible for the turntable to rotate by more than a full turn whilst the object remains above the horizon. The negative root of the quadratic applies if $|\psi| > \pi/2$, otherwise the positive root.

6 Summary and practical limitations

The effects of the Elevation and Azimuth limits and of the zenith blind spot are summarised for the WHT in Table 1. Only the Elevation limit need be considered in most cases, the possibility of encountering the Azimuth limit being restricted to a few northerly objects observed below the Pole.

Table 1: Summary of effective limits for the WHT

Declination	Effective limit
$\delta < -51.24°$	Never above Elevation limit
$-51.24° < \delta < 28.55°$	Rises and sets at Elevation limit
$28.55° < \delta < 28.97°$	Rises and sets at Elevation limit; passes through blind spot
$28.97° < \delta < 70.66°$	Rises and sets at Elevation limit
$70.66° < \delta < 71.24°$	Rises at elevation limit; sets at positive Azimuth limit (+ wrap) or elevation limit (− wrap)
$71.24° < \delta < 85.62°$	Always above elevation limit; Sets at positive Azimuth limit (+ wrap only)
$85.62° < \delta$	No limits

It should be noted that the Azimuth and Elevation limits are defined in the *mount* coordinate system and that further transformations are required in order to convert to astronomically-useful (h, δ). Since the index errors for the WHT are included to a good approximation in the encoder model, the difference between mount and topocentric apparent systems is not significant at the level of accuracy required ($\approx$ 1 minute of time). If higher precision is required, then the full pointing model must be included in the calculation. The theoretical radius of the zenith blind spot has been used, and it is likely that tracking will become progressively less accurate as this is approached. For practical purposes, a larger radius might be appropriate. The rotator limits are summarised in Table 2 (the same caveats apply as for Azimuth and Elevation limits).

Table 2: Summary of turntable limits for the WHT ($\psi = \theta - \theta_0 - \rho$, where θ is the sky position angle, θ_0 is an instrument-dependent offset and ρ is the mount position angle of the limit).

Declination	Effective limit
$\delta < -28.76^\circ$	Can hit negative mount PA limit
$-28.76^\circ < \delta < 28.76^\circ$	No limits if $\mid \sin\psi \mid > \cos\delta/0.877$ or $\mid \psi \mid > \pi/2$; otherwise can hit either mount PA limit
$28.76^\circ < \delta$	Can hit positive mount PA limit

A A FORTRAN subroutine to calculate rising and setting times

```
      SUBROUTINE TEL_LIMIT (RA, DEC, ST,
     &                  LAT, ZD_LIMIT, AZ_POS_LIMIT, AZ_VEL_LIMIT,
     &                  TEL_LIMIT_STATUS,UP,CAN_BLIND,BLINDED,
     &                  ST_RISE, ST_SET, TIME_TO_LIMIT,
     &                  BLIND_IN, BLIND_OUT, TIME_TO_BLIND)
*+
*                  +-------------------------------------+
*      La Palma    |     William Herschel Telescope      |   Royal Greenwich
*   Software Group | Roque de los Muchachos Observatory |     Observatory
*                  +-------------------------------------+
*
* PROGRAM : TEL_LIMIT
* AUTHOR  : Robert Laing                                DATE : 16-JAN-1989
*
* PURPOSE : Calculate whether a given object can hit the horizon or Azimuth
*           limits or can encounter the zenith blind spot and work out
*           corresponding rising and setting times.
*
* METHOD  : (1) Sort out the following possible cases and work out rise
*           and set times:
*
*   Description                  TEL_LIMIT_STATUS      RISE, SET (HA's)
*
*  Object always below horizon   NEVER_RISES          Unspecified
*  Telescope stopped by          EL_STOP             Refer to horizon limit
*   horizon limit
*  Telescope stopped by horizon  AZ_EL_STOP          RISE(-1), SET(-1), RISE(0)
*    limit on - wrap, azimuth                        refer to horizon limit;
*    limit on + wrap                                 SET(0) to Azimuth limit
*  Telescope stopped by azimuth  AZ_STOP             RISE(-1) etc. unspecified;
*    limit on 0 wrap; no limit                       SET(0) refers to Azimuth
*    on -1 wrap                                      limit; RISE(0) = -PI;
*                                                    TIME_TO_LIMIT only
*                                                    set if the object is
*                                                    up (concept of rising
*                                                    above the azimuth
*                                                    limit is silly)
*  Telescope never stopped by a  NO_TEL_LIMIT        Unspecified
*    limit
*
*           (2) Work out the times at which the object rises and sets,
*               subject to the above conditions, for the given sidereal
*               time.
*           (3) Determine whether the object passes through the zenith
*               blind spot and calculate times.
```

```
*
*           Note that this routine ignores the difference between mount
*           and geocentric coordinates.  This should be adequate to the
*           accuracy required (approximately 1 min of time in the
*           calculation of hour angle).  Limits are defined in mount
*           coordinates, and most of the zero-point errors are presumed
*           to be taken out in the encoder model.
*
* CHANGES : 01 RAL 03-Mar-1989 Define RISE(0) = -PI for the case where
*                              only the Azimuth+ limit applies.
*           02 RAL 26-Jun-1989 Added blind-spot calculations.
*           03 RAL 18-Feb-1991 Renamed TEL_LIMIT.  Removed calculation of
*                              apparent coordinates,as this is better done
*                              outside the routine.  Changed values of
*                              TEL_LIMIT_STATUS for consistency with turntable
*                              limit subroutine.
*           04 RAL 24-Jun-1991 All parameters now declared in DISPLAY.INC.
*
* CALLS   : sla_DRANGE, sla_DRANRM (from SLALIB)
*
*.............................................................................*
*
* INCLUDES:
*
      INCLUDE         '(_SYSINC)'
*
      INCLUDE         '(_DISDEF)'               ! Display definitions
      INCLUDE         '(_MTHDEF)'               ! Mathematical definitions
*
      INCLUDE         'DISPLAY_INC:DISPLAY.INC' ! Local parameters and
                                                ! COMMON
*
* GIVEN   : ================ Declaration of given parameters ================*
*
* Geocentric apparent coordinates
*
      DOUBLE PRECISION RA                   ! RA (rad)
      DOUBLE PRECISION DEC                  ! Dec (rad)
      DOUBLE PRECISION ST                   ! Sidereal time (rad)
      DOUBLE PRECISION LAT                  ! Latitude (rad)
      DOUBLE PRECISION ZD_LIMIT             ! Horizon limit in ZD (rad),
                                            ! approximately corrected
                                            ! to topocentric coordinates
      DOUBLE PRECISION AZ_POS_LIMIT         ! Azimuth + limit approximately
                                            ! corrected to topocentric
                                            ! coordinates (rad)
      DOUBLE PRECISION AZ_VEL_LIMIT         ! Azimuth velocity limit
                                            ! (absolute value in rad/s)
*
```

```
*==================== Declaration of returned parameters =====================*
*
      INTEGER          TEL_LIMIT_STATUS          ! Which limit will stop telescope
                                             ! first? (Coded as below)
*
* Following 4 variables are indexed on Azimuth wrap, since there are two
*  possible set times
*
      DOUBLE PRECISION ST_RISE(-1:0)         ! ST when source becomes visible
      DOUBLE PRECISION ST_SET(-1:0)          ! ST when source sets
      DOUBLE PRECISION TIME_TO_LIMIT(-1:0)   ! Sidereal time until next
                                             ! limit (rise or set)
      LOGICAL          UP(-1:0)              ! Is object currently visible?
      LOGICAL          CAN_BLIND             ! Can telescope encounter zenith
                                             ! blind spot for this object?
                                             ! (.TRUE. => yes)
*
* The following variables are only calculated if CAN_BLIND is .TRUE.
*
      LOGICAL          BLINDED               ! Is object currently passing
                                             ! through zenith blind spot?
                                             ! (.TRUE. => yes)
      DOUBLE PRECISION BLIND_IN              ! ST of entering blind spot
      DOUBLE PRECISION BLIND_OUT             ! ST of leaving blind spot
      DOUBLE PRECISION TIME_TO_BLIND         ! Sidereal time interval until
                                             ! object enters
                                             ! (if BLINDED = .FALSE.) or
                                             ! leaves (BLINDED = .TRUE.)
                                             ! the blind spot
*
*==================== Declaration of local variables =========================*
*
      DOUBLE PRECISION V                     ! Velocity in units of
                                             ! tracking rate
      DOUBLE PRECISION RISE(-1:0)            ! HA when source becomes visible
      DOUBLE PRECISION SET(-1:0)             ! HA when source sets
      DOUBLE PRECISION DEC_NEVER_RISES       ! Declination above which
                                             ! an object never drops
                                             ! below Elevation limit
      DOUBLE PRECISION DEC_NEVER_SETS        ! Dec below which an object
                                             ! never rises above the
                                             ! Elevation limit
      DOUBLE PRECISION DEC_AZ_PLUS           ! Dec above which object cannot
                                             ! hit Azimuth + limit
      DOUBLE PRECISION DEC_AZ_MINUS          ! Dec below which object hits
                                             ! horizon limit before Azimuth
                                             ! + limit
      DOUBLE PRECISION A1,B1,C1              ! Quadratic coefficients
      DOUBLE PRECISION TA2                   ! tan**2(Az_limit)
```

```
      DOUBLE PRECISION SPHI                   ! sin(latitude)
      DOUBLE PRECISION CPHI                   ! cos(latitude)
      DOUBLE PRECISION SPHI2                  ! sin**2(latitude)
      DOUBLE PRECISION CPHI2                  ! cos**2(latitude)
      DOUBLE PRECISION CZ                     ! cos(ZD_limit)
      DOUBLE PRECISION SD                     ! sin(Dec)
      DOUBLE PRECISION CD                     ! cos(Dec)
      INTEGER          I                      ! Loop counter
      DOUBLE PRECISION HA                     ! Current HA
      DOUBLE PRECISION COSHA_NORTH            ! Intermediate variables
      DOUBLE PRECISION COSHA_SOUTH            ! for blind-spot
      DOUBLE PRECISION HA_NORTH               ! calculation
      DOUBLE PRECISION HA_SOUTH
      DOUBLE PRECISION DENOM_SOUTH
      DOUBLE PRECISION DENOM_NORTH
      DOUBLE PRECISION FIRST_SOUTH
      DOUBLE PRECISION FIRST_NORTH
      DOUBLE PRECISION DISCR_SOUTH
      DOUBLE PRECISION DISCR_NORTH
      DOUBLE PRECISION HA_BLIND               ! HA of leaving blind-spot
*
*=========================== Executable code ================================*
*
* Useful intermediate variables
*
      TA2   = TAN(AZ_POS_LIMIT)**2
      CZ    = COS(ZD_LIMIT)
      SPHI  = SIN(LAT)
      CPHI  = COS(LAT)
      SPHI2 = SPHI**2
      CPHI2 = CPHI**2
      CD    = COS(DEC)
      SD    = SIN(DEC)
*
* Work out Declination zones
*
      DEC_NEVER_SETS  = PI - LAT - ZD_LIMIT
      DEC_NEVER_RISES =      LAT - ZD_LIMIT
      DEC_AZ_PLUS     = ACOS(ABS(SIN(AZ_POS_LIMIT)*CPHI))
*
      A1 = 1D0 + TA2
      B1 = -2D0 * CZ * SPHI * A1
      C1 = CZ**2 * (1D0 + TA2 * SPHI2) - CPHI2
*
      DEC_AZ_MINUS = ASIN((-B1 + SQRT(B1**2 - 4D0*A1*C1))/(2D0 * A1))
*
* Work out rise and set times.  We assume that:
*
*  DEC_AZ_PLUS  > DEC_NEVER_SETS > DEC_AZ_MINUS > DEC_NEVER_RISES
```

```
*
      IF (DEC .LT. DEC_NEVER_RISES) THEN     ! Always below horizon
        TEL_LIMIT_STATUS = NEVER_RISES               ! (ignore rise/set times)
        DO I = -1,0
          UP(I) = .FALSE.
        END DO
      ELSE IF (DEC .GT. DEC_AZ_PLUS) THEN    ! No limits at all (ignore
        TEL_LIMIT_STATUS = NO_TEL_LIMIT              ! rise and set times)
        DO I = -1,0
          UP(I) = .TRUE.
        END DO
      ELSE IF (DEC .LT. DEC_AZ_MINUS) THEN   ! Only horizon limit is
        TEL_LIMIT_STATUS = EL_STOP                   ! relevant
        DO I = -1,0
          SET(I) = ACOS((CZ-SD*SPHI)/(CD*CPHI))  ! HA of set in [0,PI]
          RISE(I) = -SET(I)                      ! HA of rise in [-PI,0]
        END DO
      ELSE                                       ! Azimuth limit on 0 wrap
        SET(0) = ACOS ((TA2*SD*SPHI*CPHI -
     &                   SQRT(TA2*(CD**2 - CPHI2) + CD**2))/
     &                   (CD*(1D0 + TA2*SPHI2)))
        IF (DEC .LT. DEC_NEVER_SETS) THEN    ! Elevation limit relevant
          DO I = -1,0
            RISE(I) = -ACOS((CZ-SD*SPHI)/(CD*CPHI))
          END DO
          SET(-1) = -RISE(-1)
          TEL_LIMIT_STATUS = AZ_EL_STOP
        ELSE                                     ! No horizon limit
          TEL_LIMIT_STATUS = AZ_STOP
          RISE(0) = -PI                          ! Not a real rise time
                                                 ! but needed to define
                                                 ! UP(0) consistently.
        END IF
      END IF
*
* Now work out whether the object is up at the specified ST.  Cases of objects
* which never rise or never set have been dealt with already.   First
* work out the current HA and range in [-PI,PI]:
*
*
      HA = sla_DRANGE(ST-RA)
*
      IF (TEL_LIMIT_STATUS .EQ. EL_STOP .OR.                 ! Rise and set times
     &    TEL_LIMIT_STATUS .EQ. AZ_EL_STOP) THEN             ! specified
        DO I = -1,0
          IF (HA .GT. RISE(I) .AND. HA .LT. SET(I)) THEN
            UP(I) = .TRUE.
            TIME_TO_LIMIT(I) = SET(I) - HA
          ELSE
```

```
          UP(I) = .FALSE.
          TIME_TO_LIMIT(I) = sla_DRANRM(RISE(I) - HA)
        END IF
      END DO
    ELSE IF (TEL_LIMIT_STATUS .EQ. AZ_STOP ) THEN
      UP(-1) = .TRUE.                                  ! Up on - wrap
        IF (HA .GT. RISE(0) .AND. HA .LT. SET(0)) THEN
          UP(0) = .TRUE.
          TIME_TO_LIMIT(0) = SET(0) - HA
        ELSE
          UP(0) = .FALSE.
        END IF
    END IF
*
* Range ST in [0,2PI] for output
*
    DO I = -1,0
      ST_RISE(I) = sla_DRANRM(RA + RISE(I))
      ST_SET(I)  = sla_DRANRM(RA + SET(I))
    END DO
*
* Blind spot calculation
*
    V = AZ_VEL_LIMIT / (15D0 * AS2RAD)  ! Velocity in units of tracking speed
*
    DISCR_NORTH  = SQRT(SD**2 + 4D0*V*SPHI + 4D0*V*V)
    DISCR_SOUTH  = SQRT(SD**2 - 4D0*V*SPHI + 4D0*V*V)
    FIRST_NORTH  = SD*(1D0+2D0*V*SPHI)
    FIRST_SOUTH  = SD*(1D0-2D0*V*SPHI)
    DENOM_SOUTH  = 2D0*V*CD*CPHI
    DENOM_NORTH  = -DENOM_SOUTH
*
* cos(HA) for entering/leaving blind spot calculated for objects S and
* N of the zenith
*
    COSHA_SOUTH  = (FIRST_SOUTH + DISCR_SOUTH)/DENOM_SOUTH
    COSHA_NORTH  = (FIRST_NORTH - DISCR_NORTH)/DENOM_NORTH
*
    IF      (ABS(COSHA_SOUTH) .LT. 1D0) THEN
      HA_BLIND   = ACOS(COSHA_SOUTH)
      CAN_BLIND = .TRUE.
    ELSE IF (ABS(COSHA_NORTH) .LT. 1D0) THEN
      HA_BLIND   = ACOS(COSHA_NORTH)
      CAN_BLIND = .TRUE.
    ELSE
      CAN_BLIND = .FALSE.
    END IF
*
    IF (CAN_BLIND) THEN
```

```
          BLIND_IN  = sla_DRANRM(RA - HA_BLIND)
          BLIND_OUT = sla_DRANRM(RA + HA_BLIND)
          IF (HA .GT. -HA_BLIND .AND. HA .LT. +HA_BLIND) THEN
            BLINDED = .TRUE.
            TIME_TO_BLIND = +HA_BLIND - HA
          ELSE
            BLINDED = .FALSE.
            TIME_TO_BLIND = sla_DRANRM(-HA_BLIND - HA)
          END IF
        END IF
*
*==============================================================================*
*
        END
```

B A FORTRAN subroutine to calculate rotator limits

```
      SUBROUTINE ROT_LIMIT (RA, DEC,
     &                            ST, LAT, SKY_PA, MOUNT_PA,
     &                            LIMIT, OFFSET,
     &                            LIMIT_STATUS, ST_LIMIT,
     &                            TIME_TO_LIMIT)
*+
*                   +------------------------------------+
*      La Palma     |     William Herschel Telescope     |   Royal Greenwich
*   Software Group  | Roque de los Muchachos Observatory |     Observatory
*                   +------------------------------------+
*
* PROGRAM : ROT_LIMIT
* AUTHOR  : Robert Laing                                DATE : 17-Feb-1991
*
* PURPOSE : Calculates whether a given object can hit one of the instrument
*           rotator limits and if so, when it does so.
*
* METHOD  : There are three possible cases:
*
*   Dec < -Lat             Will hit negative mount PA limit
*
* -Lat < Dec < Lat         No limits if |sin parallactic angle|
*                          > cos(Dec)/cos(Lat); otherwise can hit
*                          either limit.
*
*   Dec > Lat              Will hit positive mount PA limit
*
* RESTRICTIONS: This routine ignores the difference between mount
*           and geocentric coordinates.   This should be adequate to the
*           accuracy required (approximately 1 min of time in the
*           calculation of hour angle (limits are defined in mount
*           coordinates).
*
*           It is assumed that the latitude is greater than 0.
*
* CHANGES : 01 RAL 24-Jun-1991 Parameter declarations from DISPLAY.INC.
*           02 RAL 09-Aug-1991 Fix typo in calculation of TIME_TO_LIMIT.
*           03 RAL 14-Aug-1991 Major rewrite.
*           04 RAL 15-Aug-1991 Deal with case that rotator is outside
*                              potential limit zone.
*
* CALLS   : sla_DRANGE, sla_DRANRM from SLALIB.
*
*...........................................................................*
*
```

```
* INCLUDES:
*
      INCLUDE         '(_SYSINC)'               ! System includes (for SLALIB
                                                ! routines EXTERNAL
                                                ! declaration)
*
      INCLUDE         '(_DISDEF)'               ! Display definitions
      INCLUDE         '(_MTHDEF)'               ! Mathematical definitions
*
      INCLUDE         'DISPLAY_INC:DISPLAY.INC' ! Local COMMON and parameter
                                                ! declarations
*
* GIVEN   : ================= Declaration of given parameters ==============*
*
      DOUBLE PRECISION RA                    ! RA (rad)
      DOUBLE PRECISION DEC                   ! Dec (rad)
      DOUBLE PRECISION ST                    ! Sidereal time (rad)
      DOUBLE PRECISION LAT                   ! Latitude (rad)
      DOUBLE PRECISION MOUNT_PA              ! Current mount PA
      DOUBLE PRECISION LIMIT(ROT_PLUS_LIMIT:ROT_MINUS_LIMIT)
                                             ! Mount PA limits (rad)
                                             ! ROT_PLUS_LIMIT  => +ve
                                             ! ROT_MINUS_LIMIT => -ve
      DOUBLE PRECISION SKY_PA                ! Current sky position angle
      DOUBLE PRECISION OFFSET                ! Instrument offset (rad)
                                             ! This is DIS_GD_SLIT_OFF in TCS
*
*================== Declaration of returned parameters =====================*
*
      INTEGER          LIMIT_STATUS          ! Which limit will stop rotator
                                             ! first? (Coded as below)
*
* The two following variables are only valid if a limit can be hit, i.e.
* if LIMIT_STATUS is non-zero
*
      DOUBLE PRECISION ST_LIMIT              ! Sidereal time when limit is
                                             ! hit
      DOUBLE PRECISION TIME_TO_LIMIT         ! Sidereal time until next
                                             ! limit
*
*================== Declaration of local variables =========================*
*
      DOUBLE PRECISION HA                    ! Current HA
      INTEGER          TURNS                 ! Number of whole turns between
                                             ! current turntable position and
                                             ! limit
      INTEGER          I                     ! Loop counter
      INTEGER          ROOT                  ! +1 for +ve root of
                                             ! quadratic; -1 for -ve
```

```
                                          ! root
      INTEGER          SIGN               ! Sign of parallactic angle
                                          ! (and therefore hour
                                          ! angle)
      DOUBLE PRECISION HA_FIRST_LIMIT     ! Hour angle of first limit to
                                          ! be encountered
      DOUBLE PRECISION PARANG             ! Parallactic angle of
                                          ! applicable limit
      DOUBLE PRECISION Q,CH
*
* Following variables refer to + (index = ROT_PLUS_LIMIT) and
* - (index = ROT_MINUS_LIMIT) limits
*
      DOUBLE PRECISION PSI(ROT_PLUS_LIMIT:ROT_MINUS_LIMIT)
                                          ! Parallactic angles
                                          ! corresponding to
                                          ! + and - mount PA limits
      LOGICAL          EXISTS(ROT_PLUS_LIMIT:ROT_MINUS_LIMIT)
                                          ! .TRUE. if limit can be hit
      DOUBLE PRECISION HA_LIMIT(ROT_PLUS_LIMIT:ROT_MINUS_LIMIT)
                                          ! Hour angle of limit
*
*========================== Executable code ================================*
*
* Parallactic angles corresponding to the positive and negative mount PA
* limits, ranged in [-PI,PI]
*
      DO I = ROT_PLUS_LIMIT,ROT_MINUS_LIMIT
        PSI(I) = sla_DRANGE(SKY_PA - OFFSET - LIMIT(I))
      END DO
*
* Hour angle ranged in [-PI,PI]
*
      HA = sla_DRANGE(ST-RA)
*
      IF (ABS(DEC) .GE. LAT) THEN          ! |DEC| > LAT: 1 solution
        IF (DEC .GE. LAT) THEN             ! + mount PA limit
          LIMIT_STATUS = ROT_PLUS_LIMIT
          PARANG = PSI(ROT_PLUS_LIMIT)
        ELSE                               ! - mount PA limit
          LIMIT_STATUS = ROT_MINUS_LIMIT
          PARANG = PSI(ROT_MINUS_LIMIT)
        END IF
        IF (PARANG .GE. 0D0) THEN          ! + hour angle
          SIGN = +1
        ELSE                               ! - hour angle
          SIGN = -1
        END IF
        IF (ABS(PARANG-PIBY2) .LT. 1D-10 .OR.   ! Trap to prevent
```

```
     &      ABS(PARANG+PIBY2) .LT. 1D-10) THEN  ! tan(90 deg)
          IF (ABS(DEC-PIBY2) .LT. 1D-10 .OR.    ! Special case N
     &        ABS(DEC+PIBY2) .LT. 1D-10) THEN   ! and S poles
            HA_FIRST_LIMIT = SIGN * PIBY2       ! +/- 6 hours
          ELSE
            CH = TAN(LAT)/TAN(DEC)
            IF (CH .GE. 1.0D0) THEN
              HA_FIRST_LIMIT = 0D0
            ELSE IF (CH .LE. -1.0D0) THEN
              HA_FIRST_LIMIT = PI
            ELSE
              HA_FIRST_LIMIT = SIGN * ACOS(CH)
            END IF
          END IF
        ELSE                                    ! Par angle not 90 deg
          IF (ABS(PARANG) .GT. PIBY2) THEN
            IF (DEC .GE. LAT) THEN
              ROOT = +1                         ! + root of quadratic
            ELSE
              ROOT = -1                         ! - root of quadratic
            END IF
          ELSE
            IF (DEC .GE. LAT) THEN
              ROOT = -1                         ! - root of quadratic
            ELSE
              ROOT = +1                         ! + root of quadratic
            END IF
          END IF
          CALL QUADSOLVE(PARANG,DEC,LAT,SIGN,   ! Solve quadratic for
     &                   ROOT,HA_FIRST_LIMIT,   ! cos h
     &                   EXISTS(LIMIT_STATUS))
          IF (.NOT.(EXISTS(LIMIT_STATUS))) THEN ! No solution: must
            IF (ROOT .EQ. +1) THEN              ! be because DEC and LAT
              HA_FIRST_LIMIT = 0D0              ! are very nearly equal
            ELSE
              Q = (TAN(PARANG)*SIN(LAT))**2
              CH = (Q - 1D0)/(Q + 1D0)
              IF (CH .GE. 1.0D0) THEN
                HA_FIRST_LIMIT = 0D0
              ELSE IF (CH .LE. -1.0D0) THEN
                HA_FIRST_LIMIT = PI
              ELSE
                HA_FIRST_LIMIT = SIGN * ACOS(CH)
              END IF
            END IF
          END IF
*
* Additional complication for Dec > Lat or Dec < -Lat.   The rotator may
* have to go round more than 360 deg.   This is really only relevant for
```

```
* elevations sufficiently high that the telescope is not stopped either
* by the horizon limit or the Azimuth limit (Dec > 85.62 deg for the
* WHT), but is included for completeness.
* TURNS is the number of whole turns between the current
* position and the limit.
*
          IF (LIMIT_STATUS .EQ. ROT_PLUS_LIMIT) THEN
            TURNS = INT((LIMIT(ROT_PLUS_LIMIT) - MOUNT_PA)/TWOPI)
          ELSE
            TURNS = INT((MOUNT_PA - LIMIT(ROT_MINUS_LIMIT))/TWOPI)
          END IF
        END IF
      ELSE
*
* Now do the case where -Lat < Dec < Lat.  The parallactic angle
* must be in the range +/-arcsin(cos(Lat)/cos(Dec)).  In addition,
* there is an ambiguity because the parallactic angle has been
* ranged in [-PI,PI].  Eliminate clearly impossible cases first.
*
        DO I = ROT_PLUS_LIMIT,ROT_MINUS_LIMIT   ! 2 or 0 solutions
          IF (ABS(PSI(I)) .GE. PIBY2 .OR.
     &    (I .EQ. ROT_PLUS_LIMIT .AND.
     &     LIMIT(ROT_PLUS_LIMIT) - MOUNT_PA .GT. PI) .OR.
     &    (I .EQ. ROT_MINUS_LIMIT .AND.
     &     MOUNT_PA - LIMIT(ROT_MINUS_LIMIT) .GT. PI)) THEN
            EXISTS(I) = .FALSE.
          ELSE
            IF (PSI(I) .GE. 0D0) THEN           ! + hour angle
              SIGN = +1
            ELSE                                ! - hour angle
              SIGN = -1
            END IF
            IF (I .EQ. ROT_MINUS_LIMIT) THEN    ! + root corresponds to
              ROOT = +1                         ! - mount PA limit
            ELSE                                ! - root corresponds to
              ROOT = -1                         ! + mount PA limit
            END IF
            CALL QUADSOLVE(PSI(I),DEC,LAT,SIGN, ! Solve quadratic for
     &                     ROOT,HA_LIMIT(I),    ! cos h
     &                     EXISTS(I))
          END IF
        END DO
*
* Now work out which limit (if any) gets hit first
*
        IF (EXISTS(ROT_PLUS_LIMIT) .AND.        ! Both limits valid?
     &      EXISTS(ROT_MINUS_LIMIT)) THEN
          IF (HA_LIMIT(ROT_PLUS_LIMIT) .GT.
     &        HA_LIMIT(ROT_MINUS_LIMIT)) THEN
```

```
          IF (HA .LT. HA_LIMIT(ROT_MINUS_LIMIT)) THEN
            LIMIT_STATUS = ROT_MINUS_LIMIT    ! - limit first
            HA_FIRST_LIMIT = HA_LIMIT(ROT_MINUS_LIMIT)
          ELSE IF (HA .LT. HA_LIMIT(ROT_PLUS_LIMIT)) THEN
            LIMIT_STATUS = ROT_PLUS_LIMIT     ! Between - and + limits
            HA_FIRST_LIMIT = HA_LIMIT(ROT_PLUS_LIMIT)
          ELSE                                ! - limit next time round
            LIMIT_STATUS = ROT_MINUS_LIMIT
            HA_FIRST_LIMIT = HA_LIMIT(ROT_MINUS_LIMIT)
          END IF
        ELSE                                  ! - limit has larger HA
          IF (HA .LT. HA_LIMIT(ROT_PLUS_LIMIT)) THEN
            LIMIT_STATUS = ROT_PLUS_LIMIT     ! + limit first
            HA_FIRST_LIMIT = HA_LIMIT(ROT_PLUS_LIMIT)
          ELSE IF (HA .LT. HA_LIMIT(ROT_MINUS_LIMIT)) THEN
            LIMIT_STATUS = ROT_MINUS_LIMIT    ! Between + and - limits
            HA_FIRST_LIMIT = HA_LIMIT(ROT_MINUS_LIMIT)
          ELSE                                ! + limit next time round
            LIMIT_STATUS = ROT_PLUS_LIMIT
            HA_FIRST_LIMIT = HA_LIMIT(ROT_PLUS_LIMIT)
          END IF
        END IF
      ELSE IF (EXISTS(ROT_PLUS_LIMIT)) THEN   ! + limit only
        LIMIT_STATUS = ROT_PLUS_LIMIT
        HA_FIRST_LIMIT = HA_LIMIT(ROT_PLUS_LIMIT)
      ELSE IF (EXISTS(ROT_MINUS_LIMIT)) THEN  ! - limit only
        LIMIT_STATUS = ROT_MINUS_LIMIT
        HA_FIRST_LIMIT = HA_LIMIT(ROT_MINUS_LIMIT)
      ELSE                                    ! No limits
        LIMIT_STATUS = NO_ROT_LIMIT
      END IF
*
* Finally, the rotator can be no more than PI away from the limit, and
* its rotation is periodic, so:
*
      TURNS = 0
     END IF
*
* Work out time to limit and ST of limit and range for output
*
     TIME_TO_LIMIT = sla_DRANRM(HA_FIRST_LIMIT - HA) + TWOPI*TURNS
     ST_LIMIT = sla_DRANRM(RA + HA_FIRST_LIMIT)
*
*=============================================================================*
*
     END
```

```
      SUBROUTINE QUADSOLVE (PSI, DEC, LAT, SIGN, ROOT, HA, EXISTS)
*+
*                   +-------------------------------------+
*      La Palma     |     William Herschel Telescope      |   Royal Greenwich
*   Software Group  | Roque de los Muchachos Observatory |     Observatory
*                   +-------------------------------------+
*
* PROGRAM : QUADSOLVE
* AUTHOR  : Robert Laing                                DATE : 14-Aug-1991
*
* PURPOSE : Solves the quadratic equation for cos (hour angle) needed to
*           find the time at which a rotator encounters a limit.
*
*           An hour angle corresponding to either the positive or negative
*           root is returned, depending on the value of ROOT.
*           The parameter SIGN is used to decide whether the output HA
*           should be positive or negative, as there is an ambiguity in
*           the arcos function.   If the equation has no real roots, the
*           logical flag EXISTS is set to .FALSE.   The
*           the case where the value of PSI leads to an infinite
*           tangent is presumed to be trapped outside the routine.
*
* CHANGES :
*
*====================== Includes ======================================*
*
      INCLUDE '(_MTHDEF)'
*
*====================== Declaration of given variables ================*
*
      DOUBLE PRECISION PSI                    ! Parallactic angle
      DOUBLE PRECISION DEC                    ! Declination
      DOUBLE PRECISION LAT                    ! Latitude
      INTEGER          SIGN                   ! +1 for positive hour
                                              ! angles; -1 for negative.
      INTEGER          ROOT                   ! +1 for +ve root; -1
                                              ! for negative root
*
*==================== Declaration of returned variables ================*
*
      DOUBLE PRECISION HA                     ! Hour angle corresponding
                                              ! to given parallactic
                                              ! angle
      LOGICAL          EXISTS                 ! .TRUE. if the quadratic
                                              ! has real roots
*
*==================== Declaration of local variables ========================*
*
      DOUBLE PRECISION K,L,M                  ! Quadratic coefficients
```

```
      DOUBLE PRECISION DISCR                  ! Discriminant of quadratic
      DOUBLE PRECISION PSQ                    ! tan(parallactic angle)**2
      DOUBLE PRECISION SPHI                   ! sin(latitude)
      DOUBLE PRECISION CPHI                   ! cos(latitude)
      DOUBLE PRECISION SPHI2                  ! sin**2(latitude)
      DOUBLE PRECISION CPHI2                  ! cos**2(latitude)
      DOUBLE PRECISION SD                     ! sin(Dec)
      DOUBLE PRECISION CD                     ! cos(Dec)
      DOUBLE PRECISION CH                     ! cos(HA)
*
*=========================== Executable code ================================*
*
* Useful intermediate variables
*
      SPHI  = SIN(LAT)
      CPHI  = COS(LAT)
      SD    = SIN(DEC)
      CD    = COS(DEC)
*
      PSQ = (TAN(PSI))**2
*
* Quadratic coefficients
*
      K = (1D0 + PSQ*SD**2)*CPHI**2
      L = -2D0*PSQ*SD*CD*SPHI*CPHI
      M = PSQ*CD**2*SPHI**2 - CPHI**2
      DISCR = L**2 - 4D0*K*M
*
      IF (DISCR .LT. 0D0) THEN            ! No real roots
        EXISTS = .FALSE.
      ELSE
        EXISTS = .TRUE.
        CH = (-L + ROOT*SQRT(DISCR))/(2D0*K)
        IF (CH .GE. 1D0) THEN
          HA = 0D0
        ELSE IF (CH .LE. -1D0) THEN
          HA = PI
        ELSE
          HA = SIGN * ACOS(CH)
        END IF
      END IF
*
*============================================================================*
*
      END
```