

## The Douglas Argos-Filter Algorithm

### ABSTRACT:

The Douglas Argos-Filter Algorithm ingests *Argos*<sup>1</sup> satellite tracking data (*DIAG format*<sup>2</sup> only) and flags improbable locations based on user-defined distance and velocity thresholds. The algorithm writes all results to ASCII comma-delimited files for spreadsheet or GIS applications, as well as *KML 2.0*<sup>3</sup> files compatible with *Google™ Earth*<sup>4</sup>. The entire unfiltered data set is also output, so in its minimum application, the algorithm is a useful tool for parsing Argos DIAG data into a more convenient digital format.

Service Argos disseminates each location with a location class (LC) accuracy index that falls under two broad categories: *standard*<sup>5</sup> and *auxiliary*<sup>6</sup>. Standard locations (LC 3, 2, 1) have an *estimated 1-sigma error radius*<sup>7</sup> of 250, 500, and 1500 m, respectively, while accuracy of the auxiliary locations (LC 0, A, B, Z) is highly variable and undocumented by Service Argos. Auxiliary locations are largely unsuitable for interpreting animal movements without first addressing accuracy issues.

The Douglas Filter assesses plausibility of every Argos location using two different methodologies based on: 1) distances between consecutive locations; and 2) rates and bearings among consecutive movement vectors. Both filters independently move chronologically through the raw tracking data of each individual animal, evaluating 3 locations at a time. Results of both filters are written to separate output files. The first filter searches for spatiotemporal redundancy, under the premise that significant locational errors rarely occur in the same geographic locale consecutively. The user inputs a distance threshold that defines the spatial scale of "locale". The second filter constructs 2 adjoining vectors and evaluates their velocity, internal angle of intersection, and length, under the premise that significant locational errors are commonly associated with tracking signatures that create implausible movement rates or anomalously acute movement angles. User-defined thresholds for implausible velocity and suspect angles allow the user to adjust the filtering to better conform to a species' movement capability and behavior. Depending on species, one of the two filtering methods typically possesses advantages over the other. For avian species, the algorithm also produces a third output that optimally merges the two filters, since birds intermittently migrate long distances with high velocity and directionality, but during non-migratory periods they tend to occupy relatively localized nesting, molting, staging, or wintering areas.

The Douglas Argos-Filter Algorithm is available from the *USGS Alaska Science Center*<sup>8</sup>, however, the program requires a *PC-SAS*<sup>9</sup> software license for execution.

<sup>1</sup><http://www.cls.fr>

<sup>2</sup>[http://www.cls.fr/manuel/html/chap4/chap4\\_5\\_2\\_4.htm](http://www.cls.fr/manuel/html/chap4/chap4_5_2_4.htm)

<sup>3</sup>[http://earth.google.com/kml/kml\\_intro.html](http://earth.google.com/kml/kml_intro.html)

<sup>4</sup><http://earth.google.com>

<sup>5</sup>[http://www.cls.fr/manuel/html/chap2/chap2\\_3.htm#2.3.5](http://www.cls.fr/manuel/html/chap2/chap2_3.htm#2.3.5)

<sup>6</sup>[http://www.cls.fr/manuel/html/chap2/chap2\\_3.htm#2.3.6](http://www.cls.fr/manuel/html/chap2/chap2_3.htm#2.3.6)

<sup>7</sup>[http://www.cls.fr/html/argos/general/faq\\_en.html](http://www.cls.fr/html/argos/general/faq_en.html)

<sup>8</sup><http://alaska.usgs.gov>

<sup>9</sup><http://www.sas.com>

## PURPOSE :

Ingests any number of Argos DIAG files and reformats all information into ASCII comma-delimited files. Chooses the most plausible location between the ARGOS primary and alternate locations based on minimum distance from the previous chosen location. Flags implausible locations based on two different filtering methods:

- 1) Minimum Redundant Distance (**MRD**): a user-defined distance threshold for determining locational redundancy;
- 2) Distance, Angle and Rate (**DAR**): measurements that attempt to identify implausible locations based on the fact that most suspicious ARGOS locations cause an animal to incorrectly move a substantial distance and then return, resulting in a tracking-path that goes 'out-and-back' (and/or further validated by unrealistic movement rates, depending on the temporal frequency of the locations);

A hybridization of the **MRD** and **DAR** filtered results is also created. The hybrid was specifically developed for avian tracking that includes relatively high-speed, directional, migratory events. Locations that passed the **MRD** filter are retained as 'anchor points', then chronologically intervening **DAR** locations are evaluated to determine if they adhere to directional movement when compared to the vector formed by their preceding and subsequent **MRD** anchor locations.

## REQUIREMENTS :

- A) PC BASE SAS - Version 7 or later.

This SAS program requires PC SAS because it uses DOS commands to concatenate all your individual ARGOS DIAG files into a single file before ingestion. This concatenated file will be written into the same directory where your SAS program files reside, and will be named 'total.dia'. It can be considered a temporary file, which you may delete after the SAS program executes.

- B) All your Argos ASCII-format DIAG data files in ONE directory. The SAS program has been written to accept DIAG files from CD-ROM, Telnet, and/or ADS (e-mail), in both compressed and uncompressed formats.
- C) You must create an ASCII text-file that defines each of your ptt deployments. This file is comprised of IF-THEN-DO logic blocks relating each ptt-number to the respective animal upon which the ptt was deployed, and during what time interval.
- D) You must modify the SAS program 'generic\_run702.sas' to supply several user-defined thresholds that establish minimum distances, maximum rate of movement, etc., which largely dictate how conservative or liberal the filtering will be implemented. You must also define several directory pathnames (specific to your computer) that define where your input data are stored, the name and location of your deployment definition file, and the directories where results will be written.

## CAUTIONS :

If this program filters a location, it does not necessarily mean that the location was incorrect. Conversely, and more importantly, if this program does not filter a location, it does not necessarily mean the animal visited that precise location.

The underlying premise for the **MRD** filtering logic is: 'Argos does not make a significant error in the same place twice, consecutively.' In reality, bad Argos locations sometimes do occur at the same locale, consecutively, but with a low probability of occurrence. In about 2000 control relocations, I witnessed 3 consecutive pairs of locations that would have passed typical **MRD** threshold criteria. So, if you encounter suspicious results from this program - then question those results.

This program will quickly (and systematically) get about 95+% of your Argos data "cleaned-up" to your satisfaction, but you should definitely invest some QC time reviewing the results and getting any last anomalies taken care of before you make and publish biological interpretations.

You control how conservatively the filters behave through user-defined thresholds. How conservative (or liberal) the program is allowed to behave should be commensurate with the study's spatial scale-of-interpretation.

This program does not correct the inherent locational error of any single Argos location. The program strives to reduce the average error among all locations that pass the filtering criteria.



**MANDATORY USER-CHANGES TO THE SAS EXECUTABLE PROGRAM (generic\_run702.sas)**

**DIRECTORY PATH NAMES:**

Directory paths must match those on your computer. The directories must exist. DO NOT use directory names that contain 'spaces' - DOS does not recognize the 'space' character as valid, and the program will fail.

The directory containing your raw Argos DIAG files should ONLY contain DIAG data files; e.g. do not store other types of data, documents, spreadsheets, or anything else in this directory. The DIAG data must be in its original ASCII format, but it MAY contain extraneous lines such as those captured during a TELENT session (ARGOS READY/, etc.), or email-header lines from the Argos ADS.

The following statements define: 1) the full path name to your deployment definition file; 2) the directory where you installed the ArgosFilter SAS programs and subroutines; 3) the directory where all of your DIAG data reside; 4) a directory where the results will be written in ASCII format; 5) a directory where Google Earth KML files will be written; and 6) a directory where ArcView project files will be written. All directories must exist prior to running ArgosFilter. ONLY change the portion of the statement that appears in red-font below, DO NOT change anything left of the equal sign, and ensure that the terminating semicolon is not omitted.

```

*****;
*FULL path (directory AND filename) of your deployment definition file.;
*NOTE NOTE NOTE: This file MUST exist, and YOU must create it.          ;
*SEE the accompanying 'example_deployments.txt' file for documentation.;
*****;
%let deploys = c:\argos\projects\generic\sas\example_deployments.txt;

*****;
*Directory containing this SAS program and all its subroutines.          ;
*****;
%let sasroot = c:\argos\projects\generic\sas;

*****;
*Directory containing only ASCII-format Argos DIAG format data.          ;
*****;
%let diaroot = c:\argos\projects\generic\rawdiag;

*****;
*Directory where ASCII results will be saved.                            ;
*****;
%let ascroot = c:\argos\projects\generic\ascii;

*****;
*Directory where Google Earth files will be saved.                      ;
*****;
%let kmlroot = c:\argos\projects\generic\kml;

*Directory where ArcView projects will be saved.                        ;
*****;
%let arcroot = c:\argos\projects\generic\arcview;

*****;

```

**USER DEFINED PARAMETERS:**

Details about each user-defined defined parameter are presented below. These parameters control the filtering results - several are very important. You should read the documentation for each.

```
%let minoffh = 8 ;
%let maxredun = 10 ;
%let minrate = 80 ;
%let ratecoef = 20 ;

%let gmtoffst = 0. ;
%let latmin = -90. ;
%let latmax = 90. ;
%let lonmin = -180. ;
%let lonmax = 180. ;

%let r_only = 0 ;
%let r_or_a = 1 ;

%let keep_lc = 1 ;

%let keeplast = 0 ;
%let pickday = 0 ;

%let skiploc = 0 ;

%let xmigrate = 2 ;
%let xoverrun = 1.5 ;
%let xdirect = 15 ;
%let xangle = 150 ;
%let xpercent = 15 ;

%let testp_0a = 2 ;
%let testp_bz = 3 ;

%let crossval = 0 ;

%let unixdata = 0 ;
```

**DESCRIPTIONS FOR EACH USER-DEFINED PARAMETER:****MINOFFH**

**Please read carefully; most users misunderstand, and set this parameter wrong!**

*To assign a correct threshold for this parameter, ask yourself: "What is the minimum amount of time that needs to elapse without receiving any ptt transmissions to ensure that at least one duty cycle has incremented?"*

If your ptt did not have a duty cycle (eg. it was on continuously), or if the ptt was "off" for LESS time than it was "on", read the **PICKDAY** variable below first.

The SAS program attempts to group locations based on the ptt duty cycle during which they were obtained. Most ptt's attached to animals have a preprogrammed duty cycle for conserving battery life. For example, 12 hours 'on' followed by 60 hours 'off', or 6 hours 'on' followed by 22 hours 'off'. Therefore, if enough elapsed time has passed without reception of ANY Argos data, then we can assume that AT LEAST ONE duty cycle has elapsed. For example, consider a ptt that was programmed with the duty cycle below, which changes seasonally:

```
Season-1: 6h on / 18h off for 60 days
Season-2: 8h on / 64h off for 150 days
Season-3: 4h on / 96h off for remaining battery life
```

We can conclude that if there was a temporal gap between Argos overpasses (received ptt transmissions) that exceeded 8 hours during any of the 3 seasons above, that AT LEAST ONE duty cycle had transpired.

In the example above, I would suggest a value of 10 be assigned to the **MINOFFH** variable:

```
%let minoffh = 10;
```

That is, consider your ptt's duty cycle (across all seasons if applicable), ask what is the MAXIMUM on duration (units=hours), then add a couple of hours, but don't add so many hours that the value begins to approach the minimum off time (across all seasons). I add some time because some ptt models do not use real clocks, and their estimates of elapsed time vary. I chose 10 hrs in my example above because it's liberally above the maximum on time (8h) and conservatively below the minimum off time (18h).

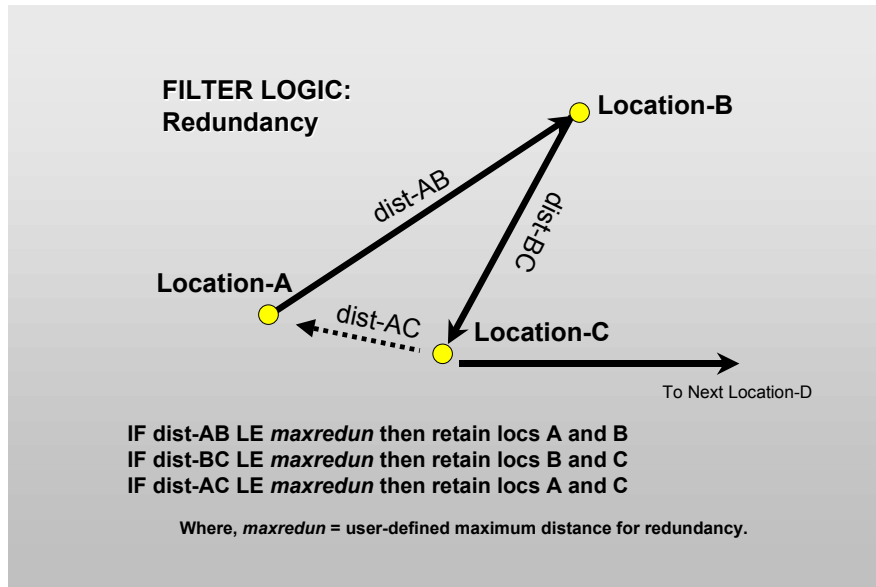
Note that if the minimum OFF duration is less than the maximum ON duration, during any season of a ptt's duty cycle, this logic fails. It will not crash the SAS program, but it will mean that the output variables PICK and DUTY will not be valid (see variable definitions below). It's less common that animal ptt's are programmed to transmit with intense battery consumption, but some studies do. If your ptt transmitted continuously (24h/day), then using the 24-hour day would be a logical choice for defining temporal intervals. See the **PICKDAY** variable below.

NOTE: The 'DUTY' variable only increments when a location is actually acquired from Service Argos. The 'DUTY' variable DOES NOT NECESSARILY EQUAL the number of 'real' elapsed duty cycles. When a ptt fails to acquire a location during any given real duty cycle, the 'DUTY' variable does not increment, and thus falls behind the actual number of expired duty cycles.

**MAXREDUN**

This variable (units=km) is used to filter the Argos locations such that every retained location has a temporally near-consecutive location that is spatially within the defined distance. The premise for such a filter is that larger Argos locational errors rarely occur consecutively in the same place (although it does happen, it is relatively rare in most datasets). The program logic is constructed as follows: The dataset is evaluated chronologically for each animal. Three consecutive locations are evaluated simultaneously - call them locations A, B, and C (see Figure below). The 3 resulting between-location distances are calculated (A->B, B->C, and C->A). If any calculated distance is less than the user-defined threshold (**MAXREDUN**), then both locations (the 2 endpoints of the vector whose length is less than **MAXREDUN**) are retained (they are not filtered). The term 'near-consecutive' is used above, because in the typical situation, location 'B' will be the outlier, and locations 'A' and 'C' will be the pair of redundant, self-confirming locations, even though they are not exactly consecutive. If location-B is flagged for omission, then should it be considered in the next set (triplet) of evaluated locations? Good question - please read the

**SKIPLOC** documentation below. If location-B was flagged for filtering but the **SKIPLOC** parameter is set to zero (0), then next set of evaluated locations would be: B->C, C->D, and D->B. If the user-defined variable **KEEPPLAST** (see below) is set to a value of one (1), then the last location for each animal will NOT be filtered, regardless of distance to the prior location.



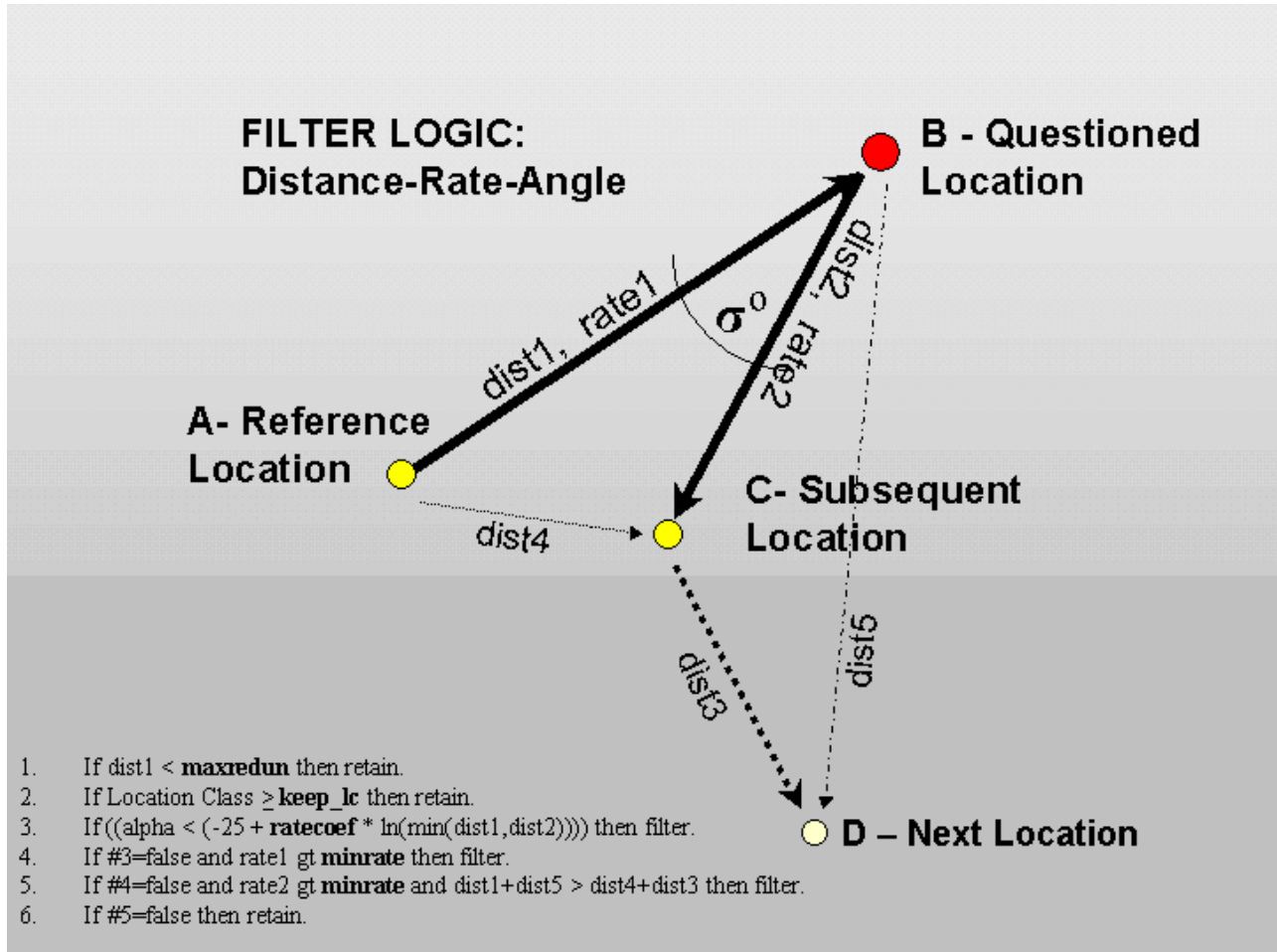
What value should be used for **MAXREDUN**? Very good question. It depends on several factors, but the most important one is 'At what spatial scale are you interpreting your location data?' For example, consider a bird tracked from Canada to Chile (very broad scale). Let's say that during migration, as the bird was travelling over northern regions of South America, 1 Argos location with a Location Class value of 'B' was obtained near the coast of Peru. If this was the ONLY location received after the bird left Cuba and arrived in Chile, then it shouldn't be concluded that this individual used a migratory pathway along the coast of Peru. A single Class-B location could have a very large error. But, what if another location of Class-B quality was obtained that was consecutive to the aforementioned location, and they were within 30km of each other? Now, 2 Argos locations, albeit of low Class-quality, have self-confirmed the same general Peruvian area. Together, they provide a lot more credibility toward concluding that the animal used the coastal region of Peru during migration to Chile. So if you're investigating continental or hemispheric-scale movements, having an animal relocated within 30km is reassuring information.

In contrast, imagine a kangaroo study, and that the study animals really didn't go anywhere - they just stayed around an area of +/- 25km. The raw Argos data are probably spread over a large geographic area, a mixture of good and implausible locations. In this situation, a very conservative **MAXREDUN** value of perhaps 2km would be more appropriate. Such conservative values will filter a lot of locations, but it may be worth having a critical look at those locations that are not filtered.

There are other factors involved with the usefulness of the **MAXREDUN** filter, such as duty cycle, latitude, and the real rate of animal movement. For example, I have been using **MAXREDUN** = 10 for filtering polar bear data. The polar bear transmitters typically transmitted for 6 or 8 hours every 3 or 7 days. During their on-period in the northern latitudes, several locations were usually obtained, especially when the ptt's were not near the end of their battery life. Because polar bears move relatively slowly, there were enough locations during the duty cycle that were within 10km of one another to provide redundancy, and the more implausible locations were fairly easy to identify. However, if you have a duty cycle that has a relatively short on-time, and/or your study area is near the equator where the frequency of satellite overpasses is minimal, then many or most of your duty cycles may have recovered only a single location. Or, if your animals are moving relatively large distances during (within) the duty cycle, your dataset will contain little locational redundancy, and the redundancy filter may not be very useful. This is typical of avian tracking when the birds are migrating. When birds are at their nesting, staging, or wintering grounds, **MAXREDUN** values of 5km or 10km may be satisfactory, but when the birds are migrating, the distance-angle-rate filter (described below) will be more useful. Merging the results of the redundancy filter and the distance-angle-rate filter, for animals like birds that nest, stage, molt, migrate, and over-winter, is also implemented in this Versions of this Program that are  $\geq 5.0$  (see **AZMARGIN**, **XMIGRATE**, and **XOVERRUN** below).

**MINRATE**

This variable (units= km/hr) is used by the DISTANCE-RATE-AZIMUTH (**DAR**) filter. It represents a measure to check plausibility of an Argos location based on a reasonable rate of movement that your animal might sustain over a period of hours or days. The program logic again assesses 3 consecutive locations simultaneously A, B, and C in chronological order (see Figure below). The program calculates 2 distances and 2 rates-of-movement, based on A->B (dist1 and rate1) and B->C (dist2 and rate2). The program also calculates the angle (alpha, in degrees) that is formed by A->B->C, with B at the vertex. Location-A is considered a valid reference location (eg. it was not previously selected for omission). Location-B is the location under question. Location C completes the triplet, but it is not considered for omission until the next iteration, when it will become the vertex-location. If location-B is flagged to be filtered, it is not used in subsequent triplets.



The program sequentially evaluates 6 tests (shown in the above figure). Note that location-B will ALWAYS be retained if the distance A->B is less than **MAXREDUN**. In other words, if the animal did not move farther than the user-defined threshold for spatial redundancy, then further tests regarding rate of movement or direction are irrelevant. Two consecutive Argos locations with good accuracy can be associated with an extremely high rate of movement when the elapsed time between the two locations is very small. Note also that if the Location Class for Location-B is greater or equal to the user-defined **KEEP\_LC** value, then Location-B is automatically retained. If the user has specified **R\_ONLY** = 1 (see below), then Test #3 is not conducted.

The third step of the assessment evaluates the angle (alpha), and permits the angle to get bigger as the distance of the smaller of the 2 vectors (dist1 or dist2) increases. If the distances are relatively small, a tighter angle is required to cause a filtered result. If the distances are relatively large, then a larger angle will cause implausibility. This is necessary, in part, because the SAS program uses great circle aviation math to derive distance and direction, but it is also biologically reasonable to presume that the farther an animal moves, the less likely it is to

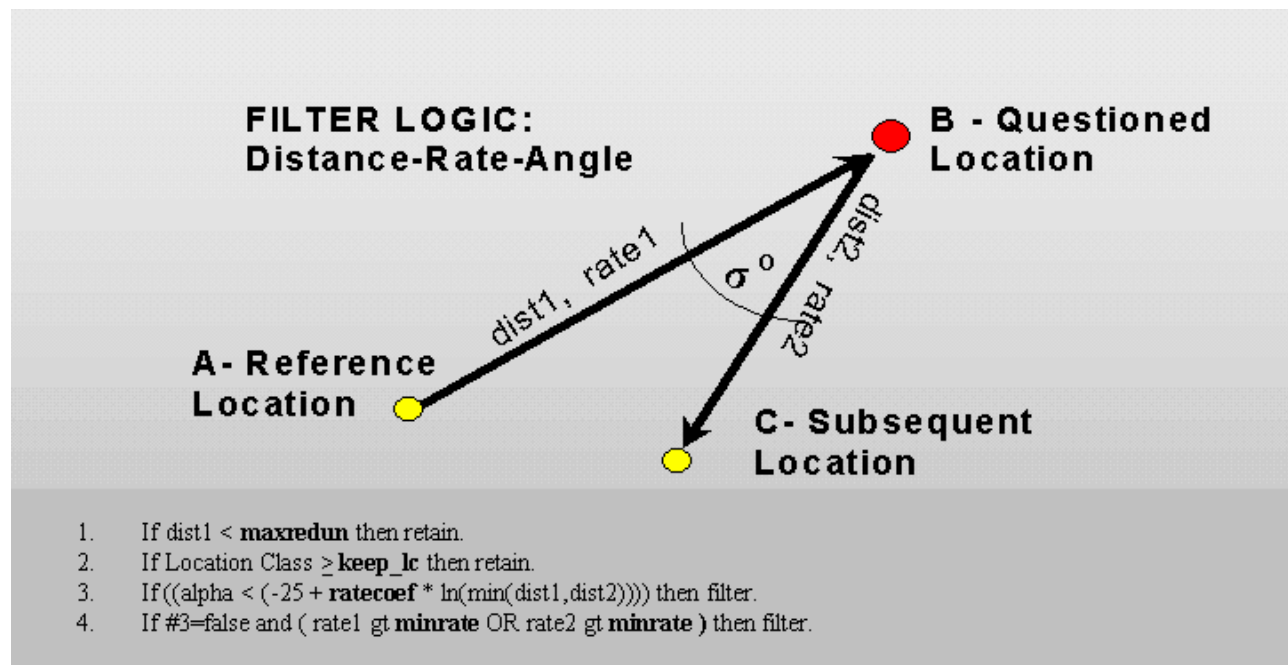


have immediately returned back toward the originating region. The user has to define a value for **RATECOEF**, which has a logarithmic influence over the behavior of the angle assessment. **RATECOEF** is discussed in more detail below.

The last part of the assessment requires that at least one of the two vectors has a rate of movement that is suspect (above the user specified minimum **MINRATE**). If **rate1** is gt **MINRATE**, then location-B is filtered. If **rate2** is gt **MINRATE** (and **rate1** lt **MINRATE**) then it becomes ambiguous whether Location-B or Location-C (or both) is implausible. Cumulative distances from Location-A to Location-D, either through Location-B or through Location-C, are compared to decide whether Location-B will be filtered.

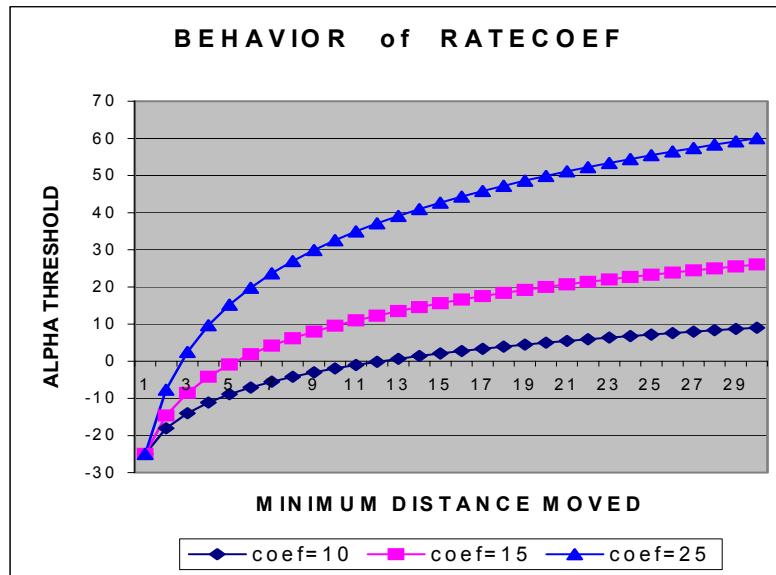
The entire **DAR** filtering strategy is iterated 3 times, each successive iteration using only locations that passed the previous iteration. It is necessary to iterate the **DAR** strategy multiple times because filtering a location during one iteration may create implausible rates of movements or suspicious angles that need to be reevaluated by a subsequent iteration.

Finally, two more **DAR** iterations are performed using a slightly different set of decision rules that are documented in the figure below. These modified rules are identical to the rules used in all previous versions of this program (e.g. Versions 5 and below). In previous program versions, the **DAR** decisions documented in the figure below were iterated 5 times. In Version 6, slightly different sets of criteria were used in the first 3 iterations to try and improve decision about the fate of Location-B when only **rate2** exceeded **MINRATE** (e.g. when it was ambiguous if Location-B, Location-C, or both were implausible).



## RATECOEF

This coefficient influences how the angle ( $\alpha$ ) is evaluated. Larger angles will be considered suspect as **RATECOEF** increases. Use of this coefficient needs further study and empirical testing. The figure below illustrates the behavior of **RATECOEF**. For example, if the shorter of the 2 vectors (A->B and B->C) was 30km and **RATECOEF**=10, then if the angle ( $\alpha$ ) was less than about 9-degrees, location B would be filtered. If **RATECOEF**=15, then angles less than about 26-degrees would be filtered, and for **RATECOEF**=25, angles less than about 60-degrees would be filtered. Assigning larger values to **RATECOEF** will result in more locations being filtered. More mobile animals should be assigned smaller **RATECOEF** values. I suggest you begin with **RATECOEF**=15 for birds, and **RATECOEF**=25 for mammals and reptiles, then review the filtering results, and adjust **RATECOEF** if you feel it's necessary or you want to experiment. I have used a **RATECOEF** value of 15 for oldsquaw, 20 for swans, and 25 for polar bears, with reasonable results. I experimented with even more aggressive (higher) **RATECOEF** values for some tracking data where the animals moved very little throughout the study. I concluded that distance-angle-rate filtering is less stable, and less intuitive for relatively sedentary animals, and that minimum-redundant-distance filtering is a better choice under such conditions.



```

r_only   = 0      (rate only)      Case 1
r_or_a   = 1      (rate or angle)  Case 2

```

One and only one choice should be flagged with a value of '1', the other must have a value of zero. The program will immediately abort if more than one choice (or no choice) is made. Filtering is made if:

In Case-1, either rate1 or rate2 must exceed the user defined **MINRATE**, without any regard to the angle formed by the two vectors.

In Case-2, either rate1 or rate2 must exceed the user defined **MINRATE** \*OR\* the angle formed by A-->B-->C must be suspiciously acute.

Most often, a user will select the 'rate or angle' (Case-2 above). A user might decide otherwise when tracking animals with particular behaviors. For example, consider sea birds such as murres or puffins that frequently travel from a colony to offshore feeding areas, and return. With this type of behavior, we might expect 'out-and-back' movements in the tracking data. The user should consider selecting the 'r\_only' choice (rate only), such that locations would only be filtered if the rate of movement exceeded **MINRATE**, regardless of the angles formed. This strategy will likely leave a substantial number of erroneous locations unfiltered. I would suggest careful review of the 'r\_only' results, searching for areas of repeated use (eg. look for temporal redundancy across days, or individuals). Repeated use of an area over time, or by multiple individuals serves to confirm, while single 'out-and-back' movements to areas that were never visited again should remain suspect.

#### **GMTOFFST (deviation from GMT, in units=hours)**

You may change all date/time values in the output data sets to have values equal to 'local' time, rather than the GMT date/time that Argos disseminates. Unit for the GMT offset variable is hours. For example, tracking conducted in California might set **GMTOFFST** = -8. If **GMTOFFST** is set to zero, then all date/time values are left unchanged (GMT). If you use a value other than zero, it may become problematic to merge or compare the output data sets to the raw Argos data, or to other data summaries. Alternatively, it is easier to interpret diurnal animal behaviors in units resembling local solar time. If your animals moved extensively between time-zones (large east-west movements), it's best to keep **GMTOFFST** = 0.

**GMTOFFST** does not adjust the DEPLOYMENT Date/Time (dtime). Thus, it is important that the dtime variable (in the deployment logic-blocks) uses the same time zone defined by **GMTOFFST**.

If **GMTOFFST** is set to 0 (zero) then all date/time values in the output data sets will be the same as those in the raw Argos data. If **GMTOFFST** is not equal to zero, the GMT\_DATE, GMT\_HOUR, GMT\_MIN,

and GMT\_SEC in the output data sets will continue to adhere to GMT time, while all other date/time values will have the offset applied.

```
latmin   =  -90.  ;
latmax   =   90.  ;
lonmin   = -180.  ;
lonmax   =  180.  ;
```

These parameters form a rectangular geographic area-of-validity. Restricting this area of validity can sometimes be useful if you have received very aberrant locations in your raw Argos data. If a location falls outside the geographic rectangle, the record is simply removed from all outputs.

### KEEP\_LC

This variable must be set to either 3, 2, 1, any other value will cause the program to fail. This parameter establishes the minimum Argos Location Class that will NEVER be filtered by either the Minimum Redundant Distance (MRD) Filter or the Rate Angle Distance (DAR) filter. The value you choose should be commensurate with the spatial scale of data analysis. We have very rarely observed an Argos Location Class 3 to have accuracy worse than about 1km (usually they are within 0.5km). However, if you are studying continental or broad regional movements, then the larger location errors associated with LC2 and LC1 (+/- 5 or 10 km respectively) may be negligible compared to the overall movements you are studying. If this is the case, then you should consider keeping all LC1, LC2 and LC3 locations (prevent them from being filtered) - thus set **KEEP\_LC** = 1. If you are studying more local scale movements, set the **KEEP\_LC** value to 3, and let redundancy self-validate the Argos data. If you are studying broad-scale movements, use **KEEP\_LC** = 1.

### RANKMETH

This variable allows the user to choose between 3 different equations that determine which location (within a duty cycle) will be identified as the 'highest quality', and carried forward into the 'Pick' (pk) datasets. The 'highest quality' location per duty-cycle is chosen based on the Argos Location Class (3,2,1,0,A,B,Z), and in the event of a tie, then the **RANKMETH** value will determine which equation is implemented to resolve the ambiguity. Valid **RANKMETH** values are 1, 2, or 3, where the value identifies one of three equations to resolve which location will be selected for the 'pick' datasets, GIVEN that the LC values are a TIE:

**rankmeth = 1:** The location with the highest IQX will prevail, else if a tie, the location with the highest IQY will prevail, else if a tie, the location with the most number of messages. This is equivalent to the method used in Versions 6.0-6.4.

**rankmeth = 2:** The location with the highest IQX will prevail, else if a tie, the location with the most number of messages will prevail, else if a tie, the location with the highest IQY. I personally lean toward this option, at this time. If you feel otherwise, I'd like to hear.

**rankmeth = 3:** The location with the most number of messages will prevail. This is equivalent to the method used in Versions 5 and earlier.

### KEEPLAST

Set this variable to 1 (one) if you want to retain the last location for each animal, within the minimum-redundant-distance filter, regardless of its location. Set the variable **KEEPLAST** to zero if you want the last location for each animal to be filtered if it is greater than **MAXREDUN** from the previous location. Recall that the last location for each animal is NEVER filtered by the distance-angle-rate (DAR) method.

### PICKDAY

Set this variable to 1 (one) if you want to select the location with the best Argos location class based on 24-hour periods instead of the ptt's duty cycle. While duty-cycle is typically a better interval from which to pick the 'best' Argos location, sometimes a biologist builds ptt's that do not have duty-cycles, e.g. the ptt is on continuously. In this case, by setting **PICKDAY** = 1, the location with the best Argos Class will be selected based 24-hour periods (eg. daily).

Important Note: If you use **PICKDAY** = 1, then set **MINOFFH** = 0.00001;

**SKIPLOC**

Consider 5 consecutive locations, and call them A, B, C, D, and E. The minimum-redundant-distance-filter considers consecutive triplets of locations; the first triplet would then be A, B, and C. Three distances are calculated: AB, BC, and AC. If any of the 3 distances are less than **MAXREDUN**, the two respective end-point locations are retained by the filter. A question arises in the event that location 'B' is rejected: should location 'B' be considered in the next triplet? If **SKIPLOC=0**, then location 'B' is considered, and the next triplet would be B, C, D; and the 3 distances BC, CD, AND BD would be evaluated. If **SKIPLOC=1**, then location 'B' is not considered (it is skipped), and the next triplet would be C, D, E; and the 3 distances CD, DE, and CE would be evaluated. Most feedback I've had indicates that users prefer **SKIPLOC=0**.

**XMIGRATE, XOVERRUN**

These variables are used to combine the minimum redundant distance filter results (**MRD**) with the distance angle rate filter results (**DAR**). When an animal is actively migrating, locational redundancy is difficult or impossible to obtain. Often, locations during active migration only pass the **DAR** filter. Beginning with Version 4.0 of this program, datasets are created that pool results from both filters. The logic assumes the **MRD** results are superior, and considers those locations that passed the **MRD** filter to be 'anchor' points. Ultimately, the merging routine supplements the **MRD** results with locations that passed the **DAR** filter, ONLY IF the **DAR** locations pass a user-defined number of directional movement tests (described in the next section). The user-defined variables **XMIGRATE**, and **XOVERRUN** control whether a **MRD** vector (a vector based on two consecutive **MRD** anchor locations) represents a 'Migration Event'. If it represents a migration event, intervening **DAR** locations collected during that vector-period are evaluated and potentially amended to the output data set, if those intervening **DAR** locations adhere to the directional movement defined by the **MRD** vector. The 'Migration Event' test for a **MRD**-vector is passed if:

1. The **MRD**-vector length exceeds **XMIGRATE\*MAXREDUN**; and
2. The intervening **DAR** vector does not exceed the **MRD** vector length + **XOVERRUN\*MAXREDUN**

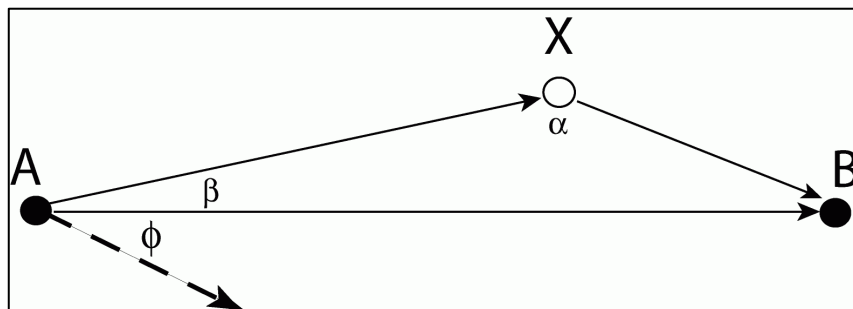
Thus, intervening **DAR** locations are not considered if the distance between two consecutive **MRD** locations was not representative of a 'Migration Event'. The distance defining a migration event is calculated as: **XMIGRATE\*MAXREDUN**. Therefore, if you had defined a relative small value for **MAXREDUN**, then the **XMIGRATE** factor would have to be adjusted larger in order to maintain the same threshold for defining a migratory movement.

The parameter **XOVERRUN** prevents a **DAR** location from being retained if that location results in a vector which exceeds the distance of the original **MRD** vector by an amount of **XOVERRUN\*MAXREDUN**. In other words, it attempts to prevent a bird from over-flying the **MRD** endpoint 'anchor' destination.

**XDIRECT, XANGLE, XPERCENT**

These variables are used when combining the minimum-redundant-distance filter results (**MRD**) with the distance-angle-rate filter results (**DAR**). These 3 parameters are used to make 3 independent tests of directionality. Using the **MRD** locations as anchor points, EACH intervening **DAR** location is INDEPENDENTLY assessed, as described below. Assuming that the **XMIGRATE** and **XOVERRUN** tests (above) were passed, 3 directionality tests are conducted: **DIRECTION**, **ANGLE**, and **PERCENT**.

In the figure below, consider two consecutive locations (A and B) that passed the **MRD** filter, and therefore have become anchor points. An intervening location (X) had passed the **DAR** filter, and will now be considered for amendment to the **MRD** results. Acceptance or rejection is based on results of three directionality tests.



TEST #1 - DIRECTION:

If the azimuth of A->X is within +/- **XDIRECT**-degrees (a user-defined threshold), location X passes the direction test. Units for **XDIRECT** are degrees, and the threshold in the figure above is illustrated by the angle  $\phi$ . In other words, if the azimuth of the vector A->X is within +/- **XDIRECT** degrees of the azimuth of A->B, then the direction test is passed.

TEST #2 - ANGLE:

This test evaluates the angle formed by A->X->B (angle  $\alpha$  in the figure above). If **XANGLE** is greater than or equal to  $\alpha$ , location X passes the angle test. **XANGLE** should not exceed 180.

TEST #3 - PERCENT DISTANCE:

If the total distance of travel along A->X->B is no greater than a user-defined percent (**XPERCENT**) of the anchor-distance A->B, then X passes the PERCENT DISTANCE test. That is, how much farther are you willing to let your animal 'wander' in order to pass through location X on its way to location B: 10%, 25%, 50%?

**TESTP\_0A & TESTP\_BZ**

**TESTPASS** defines how many of the 3 tests above (1, 2, or 3) must be passed in order for the location in question (X in the Figure above) to be retained in the 'Best Hybrid' output data set. If **TESTPASS** is defined as '1' (the most liberal), then only one (any one) of the three tests is required to pass. If **TESTPASS** is set equal to 2, then any two of the 3 tests must pass, and if **TESTPASS** is set equal to three (the most conservative), all 3 tests must pass.

Beginning in Version 6.5, **TESTPASS** was partitioned into two parts: 1) cases where the location in question is an LC=Z or LC=B; and cases where the location in question was an LC=A, LC=0, or better. If both **TESTP\_0a** and **TESTP\_bz** are set to the same value, then the net result is no different than the **TESTPASS** parameter in versions preceding 6.5.

If the user wants to be more conservative with LC=Z and LC=B locations, then the user might assign, for example:

```
%let testp_0a = 2 ;
%let testp_bz = 3 ;
```

The above requires that LC B and Z locations pass all three directionality tests in order to be retained in the 'Best Hybrid' output, while LC A and 0 locations need only pass 2 directionality tests.

If the user has defined **KEEP\_LC** = 2 or **KEEP\_LC** = 3, then the **TESTP\_0a** threshold also applies to number of affirmative directionality tests of any intervening LC1, or any LC1 and LC2 locations, respectively.

**CROSSVAL**

At the beginning of this program, a choice is made between the Argos primary and alternate locations using the entire data set of deployed ptt's. Initially, the choice is made based on sequential minimum distance criteria starting with the deployment location. When **CROSSVAL** = 0, no further evaluation is made, and the SWITCH variable in the output data sets will be either 0 (primary location chosen) or 1 (alternate location chosen). But if you are tracking a highly mobile species (e.g. a migrating bird), then the animal can sometimes travel a great distance between locations (especially between duty cycles), and the "alternate" Argos solution is more likely to fall between the actual originating and destination locations and be incorrectly chosen in the first iteration. When **CROSSVAL** = 1, the program attempts to cross-validate its initial choice between the 2 solutions by evaluating the pair of locations from the subsequent overpass. The SWITCH variable can then contain a value of: 3, if the original choice to use the alternate location was reverted back to the primary; or 4, if the original choice to use the primary location was reverted to the alternate. Until program version 7.02, the condition was always **CROSSVAL** = 1 (internal default). But, if you are tracking a species that does not move great distances (perhaps several hundreds of kilometers) between locations (especially between duty cycles if applicable), then using **CROSSVAL** = 0 is probably more stable.

Recommendation: **CROSSVAL** = 0; (slow moving animals)  
**CROSSVAL** = 1; (distances between locations is sometimes several hundreds of km)

### **UNIXDATA**

This parameter is set equal to 1, if and only if, you have received UNIX-format SAS datasets of your Argos data from David Douglas. Otherwise, it should remain equal to 0 (zero). Most users should ignore any references to UNIX and the parameter **UNIXDATA**. If you are a USGS Alaska Science Center biologist, you should coordinate with David Douglas to obtain UNIX-format SAS datasets from this program that were derived from our CD-ROM archive, rather than using DIAG data obtained through TELNET or ADS.

**OUTPUT VARIABLE DEFINITIONS:**

The table below defines each of the variables that appear in the ASCII output files. Not all variables appear in each of the ASCII files that are created. "(DIAG)" denotes a parameter that is directly copied from the original Argos DIAG input data.

Name	Format	DESCRIPTION
altitude	Num	Argos ptt reference altitude (DIAG)
animal	Char	User defined Animal ID
apbest	Num	Best signal strength during pass (DIAG)
apfreq	Num	Interpolated freq at Doppler inflection (DIAG)
azimuth	Num	Great-circle compass bearing (degrees) to next loc
azfrmdply	Num	Great-circle compass bearing (degrees) from the deployment location
bothtest	Char	Which 3 directionality tests passed in 'Best Hybrid'
date	MMDDYY	Date of overpass/location (GMT + (GMTOFFSET))
day	Num	Calendar day of overpass/location (GMT + (GMTOFFSET))
daysdply	Num	Number of elapsed days since deployment date
ddmm	Char	DAY MONTH character string
deltahr	Num	Number of elapsed hours since previous pass
deployed	Char	Character date of deployment
deplylat	Num	User declared deployment latitude
deplylon	Num	User declared deployment longitude
distance	Num	Great-circle distance (km) from previous location
dist2dply	Num	Great-circle distance (km) from deployment location
dur	Num	Duration (sec) between first and last message (DIAG)
duty	Num	Unique number for all records within a duty cycle
dutylocs	Num	Total number of passes/locations within duty cycle
first	Num	1/0 flag for first data record for a given animal
gmt date	MMDDYY	GMT date appearing in the raw Argos data (DIAG)
gmt hour	Num	GMT hour appearing in the raw Argos data (DIAG)
gmt min	Num	GMT minute appearing in the raw Argos data (DIAG)
gmt sec	Num	GMT second appearing in the raw Argos data (DIAG)
gt120	Num	N messages with signal strength <120dB (DIAG)
hour	Num	Hour of overpass/location (GMT + (GMTOFFSET))
iqx	Num	IQX value (DIAG)
iqy	Num	IQY value (DIAG)
jd	Num	Julian day YYDDD (GMT + (GMTOFFSET))
jday	Num	Julian day DDD (GMT + (GMTOFFSET))
latitude	Num	Chosen lat (1° vs 2°) based on minimum distance
lat a	Num	Alternate latitude (DIAG)
lat p	Num	Primary latitude (DIAG)
latest	Num	1/0 flag for most recent data record for an animal
lc94	Char	Argos location class
line	Char	Unique LINE-Identifier (ANIMAL-ID  MMDDYYYY  : HHMMSS  Z99)
longitud	Num	Chosen lon (1° vs 2°) based on minimum distance
lon a	Num	Alternate longitude (DIAG)
lon p	Num	Primary longitude (DIAG)
minute	Num	Minute of overpass/location (GMT) (DIAG)
mmdd	Char	MONTH DAY character string
month	Num	Month of overpass/location (GMT + (GMTOFFSET)) (DIAG)
nlocrank	Num	Weighting for best LC94 within duty cycle
nmess	Num	Number of messages received during overpass (DIAG)
nopc	Num	Number of plausibility checks passed (DIAG)
pick	Num	1/0 flag denoting best LC94 within duty cycle
ptt	Num	Ptt number (DIAG)
rate	Num	Rate of movement (km/hr) from previous loc
sampsize	Num	Total number of records for respective animal
second	Num	Second of overpass/location (GMT) (DIAG)
sen1	Char	First sensor entry from first line of data (DIAG)
sen2	Char	Second sensor entry from first line of data (DIAG)
sen3	Char	Third sensor entry from first line of data (DIAG)
sen4	Char	Fourth sensor entry from first line of data (DIAG)
switch	Num	Location Choice: 0=primary 1=alternate 3=primary 4=alternate
weeddist	Num	1/0 flag 1= reject based on minimum redundancy MRD
weedrate	Num	1/0 flag 1= reject based on distance-angle-rate DAR
weebboth	Num	1/0 flag 1= reject amending DAR loc with MRD anchors
year	Num	Year of overpass/location (GMT + (GMTOFFSET)) (DIAG)

**OUTPUT FILE DEFINITIONS:**

The SAS program automatically generates the "filename" and embeds 3 of the more critical user-defined filtering parameters (numbers), so results derived with different parameter choices can be later compared.

d# - your choice for the minimum-redundant-distance (km)  
 r# - your choice for maximum sustainable rate of movement (km/hr)  
 lc# - your choice for the best Argos location class to ALWAYS retain

The ASCII files are named with the following format, and they contain:

- tottot.txt** - One record for every DIAG overpass, regardless if a location was reported. This file provides a complete dataset of ALL Argos records. It is suited to studying ptt performance; it is not applicable for GIS since location data may be missing for many records.
- totloc\_d#\_r#\_lc#.txt** - One record for every DIAG overpass that fixed a location. Each location record is flagged with three pass-fail 1/0 variables, one for each filtering method.
- d#lc#.txt** - Subset of totloc\*.txt which contains ONLY the locations that passed the minimum-redundant-distance (**MRD**) plausibility filter - plus ALL Class-lc# and above locations. This subset contains NEW calculations of DISTANCE, RATE and AZIMUTH variables.
- r#d#lc#.txt** - Subset of totloc\*.txt which contains ONLY the locations that passed the distance, angle, rate (**DAR**) plausibility filter - plus ALL Class-lc# and above locations. This subset contains NEW calculations of DISTANCE, RATE and AZIMUTH variables.
- br#d#lc#.txt** - Locations in d#lc# PLUS additional locations from the r#d#lc# data set that conformed to the overall directional movement defined by the locations in dt#lc#. This is the pooled **MRD** and **DAR** dataset (the "best" hybrid). A few of the original **MRD** locations (anchor points) may be removed because a final movement-rate filter is applied to this data set.
- pkd#lc#.txt** - Subset of d#lc#.txt which contains ONLY the locations that were selected as the 'highest quality of the duty-cycle'. The 'highest quality' location per duty-cycle is chosen based on the Argos Location Class (3,2,1,0,A,B,Z), and in the event of a tie, the RANKMETH user-defined variable (see below) will determine which location is chosen. The "pick" (pk\*) subsets contain NEW calculations of the DISTANCE, RATE and AZIMUTH variables.
- pkrd#lc#.txt** - Subset of r#d#lc#.txt which contains ONLY the locations that were selected as the 'highest quality of the duty-cycle'. The 'highest quality' location per duty-cycle is chosen based on the Argos Location Class, as described above.
- pkbr#d#lc#.txt** - Subset of br#d#lc#.txt which contains ONLY the locations that were selected as the 'highest quality of the duty-cycle'. The 'highest quality' location per duty-cycle is chosen based on the Argos Location Class, as described above.

NOTE: It is the user's decision which filtering method (e.g. output dataset) best suits their analytical requirements. The **MRD** filter is often the best choice if tracking a slow moving animal like a mammal. Avian tracking during active migration poses challenges to the **MRD** filter because spatial redundancy is very difficult to obtain. The **DAR** filter often performs better during active migration, while the **MRD** filter is superior during sedentary periods such as nesting, staging, molting, and migratory stopovers. The "best" hybrid filter is an attempt to maintain the robustness of the **MRD** filter during sedentary periods, and use **DAR** results only during migratory periods. Keep in mind that **DAR** locations retained in the "best hybrid" data sets are those that passed directional movement tests. Consequently, we have imposed a constraint (bias) that the animal migrates in a more or less 'straight line'. But, how correct is this constraint? The 3 respective output subsets (files beginning with the letter 'p') contain just one location per duty cycle (or one location per 24h if the ptt is continuously on), and these datasets are best used for spatial analyses like home range because they reduce problems associated with temporal autocorrelation.



**THINGS YOU SHOULD KNOW ABOUT:**

01. The program never filters Argos Class-3 locations. If the user defined variable **KEEP\_LC** is set to 2 (two), then the program never filters Class-2 or Class-3 locations. If **KEEP\_LC** is set to 1 (one), then the program never filters Class-1,2,or 3 locations. If the user defines **MAXREDUN**  $\geq 10$  km, then it would make sense to set **KEEP\_LC** = 1, since Class 1, 2, and 3 locations usually have accuracy within 10 km (e.g. there is no reason to filter the Standard Location Class data).
02. In the DISTANCE-ANGLE-RATE (**DAR**) filter, the last location for each animal is NEVER filtered. In the MINIMUM-REDUNDANT-DISTANCE (**MRD**) filter, the last location for each animal may or may not be filtered, depending on its proximity to the previous location and the value you assign to the **KEEPLAST** parameter.
03. If an animal has ONLY 1 or 2 locations in the entire data set, no filtering is attempted.
04. If you are interested in a detailed analysis of ptt sensor data, this is not the program to use. The Argos DISPOSE file format contains your full set of sensor data; the DIAG format only contains ONE message of sensor data per overpass which simply serves as a solitary 'example' of your sensor data.
05. If you study the SAS program code, you should know that goodlat=latitude and goodlon=longitud. Most of the program code references goodlat/goodlon, but I changed the variable names at the end of the program to accommodate default entry into ARC/VIEW through TABLES.
06. Don't be alarmed by the series of 'INVALID DATA' errors that will appear in the SAS Log after reading in the input DIAG data file. The errors arise when the program tries to read a lat/lon in a DIAG overpass that only has '????' reported the lat/lon fields (No Argos location was reported). In this case, lat and lon are set to SAS missing values, which is correct.
07. CAUTION: The output ASCII files contain ONE DEPLOYMENT record for each animal. This was necessary to correctly choose and filter the very first ARGOS location. However, most variables for this deployment record are invalid, they are set to dummy values or they are just duplicate values of the first REAL Argos data for the respective animal. Before quantifying metrics about the ARGOS data, be sure to remove the deployment records. In the deployment records, the variable LC94 equals 'DP', so you can use this variable to selectively remove the deployment records.
08. The variable named 'switch' has a more detailed interpretation because I added code to re-evaluate the initial choice between the Argos primary and alternate locations that was based solely on minimum travel distance. The new code considers where the animal is located one overpass subsequent to the overpass being evaluated. I found that choices were more correct using this additional logic. Now, 'switch' has four categories:
 

switch = 0	Original choice of the primary location also passed 2nd test.
switch = 1	Original choice of the alternate loc also passed the 2nd test.
switch = 3	Original choice of the alternate loc was reverted to primary in the 2nd test.
switch = 4	Original choice of the primary loc was reverted to alternate in the 2nd test.
09. The choice of the best quality Argos location per duty cycle (variable 'pick') may vary between the output ASCII files that underwent filtering, because the location that was picked in one data set may have been filtered in another data set, e.g. the variable 'pick' is chosen independently for each data set.
10. You may find it necessary to 'override' this program's choice between the primary and alternate Argos location for some specific overpasses. This situation happens more frequently when an animal is moving large distances between duty cycles, and/or the ptt is near the end of its battery life. There is an accompanying ASCII text file named "hardcode\_changes.txt" in which you can: 1) force the primary location to be chosen; 2) force the alternate location to be chosen; or 3) remove the entire overpass from all output data sets. The "hardcode\_changes.txt" file contains further documentation. Obviously, you need to study the initial filtered results in a GIS (ArcView) to discover any aberrant overpasses that would benefit by forcing the choice between the primary and alternation location (or removing the overpass entirely). The examples I provide are commented, you must 'uncomment' the SAS code by removing the preceding \* (asterisk) in column #1 of each line. Also, be very careful when you edit the "hardcode\_changes.txt" file - ONLY USE an ASCII editor, and DO NOT CHANGE the file's name: it must be named "hardcode\_changes.txt".

Example of 3 strategies using logic blocks to (respectively): 1) force selection of the alternate Argos location; 2) force selection of the Argos primary location; and 3) completely remove the overpass from all output data sets.

```
* if line = 'bhg0101162000:224848Z99' then do;
*   lat_p = lat_a;
*   lon_p = lon_a;
*   file log; put 'Forced alternate location at: ' line= ;
* end;

* if line = 'bhg0303182000:201439Z99' then do;
*   lat_a = lat_p;
*   lon_a = lon_p;
*   file log; put 'Forced primary location at: ' line= ;
* end;

* if line = 'bhg0303182000:201439Z99' then do;
*   return;
*   file log; put 'Removed an entire Argos record at: ' line= ;
* end;
```

---

My contact information, should you have any questions or suggestions:

David Douglas  
USGS Alaska Science Center  
3100 National Park Road  
Juneau, AK 99801

907-364-1576 voice

[david\\_douglas@usgs.gov](mailto:david_douglas@usgs.gov)

```
*****;
* HISTORY OF PROGRAM MODIFICATIONS ;
* *****;
```

Version	Date
V2.1	April 2000
V2.2	July 2000
V2.3	August 2000
V2.4	February 2001
V2.5	February 2001
V3.1	September 2001
V4.0	November 2001
V4.1	November 2001
V5.0	July 2002
V5.01	October 2002
V5.02	January 2003
V6.0	April 2003
V6.1	August 2003
V6.2	September 2003
V6.3-4	October 2004
V6.5	January 2005
V6.6	November 2005
V6.7	January 2006
V6.8	March 2006
V7.0	April 2006
V7.02	June 2006

```
*****;
* MODIFIED Feb 4, 2000 to skip filtered locations during the minimum distance ;
* redundancy test. ;
* ;
* Also, added 2 more output datasets: ;
* 1. outdis3 = minimum redundant distance filter + all LC=3 locations ;
* 2. outpik = the best pick per duty cycle from #1 above. ;
* ;
* Also, does NOT necessarily keep the last location in redundancy testing;
* *****;
* *****;
* MODIFIED Feb 5, 2000 ;
* ;
* Requires the deployment lat/lon to be entered with the animal ID ;
* Always keeps the deployment record in the output datasets. ;
* ;
* User can pass a GMT offset parameter for correcting to local time ;
* in the USER portion of the program. ;
* ;
* For the Deployment Location, nlocrank is set to 800.00 so that it ;
* will always be PICKED as the representative site for the 1st duty cycle.;
* ;
* *****;
* *****;
* MODIFIED Feb 6, 2000 ;
* ;
* User input for area of interest lat/lon rectangle ;
* ;
* Macro for deployment information ;
* ;
* A lot of the header program was moved into the generic portion ;
* ;
* Calls DAR calculations through MACROS ;
* Requires a SAS program: calcdist.sas ;
* ;
* Always RETAINS locations w/ lc94='L3' in the Dist-Angle-Rate DAR filter;
* Always RETAINS locations w/ lc94='L3' in the Min Redundant Dist filter ;
* ;
* New RATECOEF User Parameter that adjusts the DAR selection function. ;
* *****;
* *****;
```

```

* MODIFIED July 21, 2000 ;
* ;
* User defined variable keep_lc = 3,2,1, or 0 that defines LC classes ;
* (minimum) that will never be filtered - MIN REDUN FILTER ONLY ;
* the Dist-Rate-Angle filter still keeps LC3, but filters any other LCs ;
* ;
* Disabled the skiploc function in the MIN REDUN filter, eg. if a loc is ;
* filtered it is still used as point A in the next triplet. ;
* ;
* ***** ;
* ***** ;
* MODIFIED July 31, 2000 ;
* ;
* Adjusts deployment times based on the users GMTOFFSET value ;
* ;
* ***** ;
* MODIFIED August 3, 2000 ;
* ;
* KEEP_LC is now applied to both the filters, Redundancy and DAR. ;
* ;
* ***** ;
* MODIFIED December 26, 2000 ;
* ;
* DIAG Ingestion now accepts the Argos Compressed format for ADS. ;
* Variables: test4 test5 ;
* May have missing variables because it is not a fixed format file. ;
* ***** ;
* MODIFIED January 27, 2001 ;
* ;
* Includes ddmm output variable, DAY/MONTH in $5 format. ;
* New user variable: pickday, to be used when ptt doesn't have a duty ;
* cycle, set pickday=1, and best location per calendar day is selected. ;
* ***** ;
* MODIFIED November 7, 2001 ;
* ;
* New user variable: skiploc, when set to a value of 1 it forces the ;
* redundant distance filter to skip a rejected location, thus not ;
* allowing to become point #1 of the next triplet. Otherwise set = 0. ;
* ;
* Names totloc.dbf with additional info specific to the maxrate and ;
* redundant distance user-options. ;
* ***** ;
* ***** ;
* MODIFIED November 9, 2001 Version 4.0 developed ;
* ;
* New pair of output dbf files: merged DAR and DIST filter results. ;
* User must define 2 new input variables: azmargin and xmigrate. ;
* azmargin is +/- degrees of tolerance to accept for azimuth deviations. ;
* xmigrate is integer scaling of maxredun that defines a migration event ;
* ;
* ***** ;
* MODIFIED November 15, 2001 Version 4.1 developed ;
* ;
* Optionally selects between DIAG input, or UNIX exported SAS dataset ;
* with the user defined variable UNIXDATA. ;
* ;
* ***** ;
* ***** ;
* MODIFIED May 13, 2002 Version 5.0 developed ;
* ;
* Outputs comma-delimited ASCII files instead of dBase files. ;
* ;
* ;
* ***** ;
* ***** ;
* MODIFIED July 16, 2002 Version 5.0 revised ;
* ;
* Does a better job of removing redundant overpasses. ;
* LINE, LAT_P, and LON_P must be equivalent to define redundancy ;
* ;
* ***** ;

```

```

* *****;
* MODIFIED October 25, 2002 Version 5.01 developed ;
* ;
* Assigns duty cycle = 0 to the DP deployment location. ;
* As such, the deployment location is always picked AND the best ;
* location of the first real duty cycle will also be picked. ;
* ;
* Deployment date-time variables are now corrected. ;
* Before, they were being assigned the date-time of the first overpass. ;
* *****;
* *****;
* MODIFIED January 30, 2003 Version 6.00 developed ;
* ;
* After the original MRD filter is executed, the results are subjected ;
* to a RATE-ONLY filter to remove an instances when redundancy was ;
* observed, but in fact the rate discounts the plausibility. After the ;
* RATE-ONLY is executed, the MRD is re-executed to ensure redundancy. ;
* ;
* The strategy for pooling the MRD and DAR results was completely ;
* restructured. Using the MRD locations as anchor points, it assesses ;
* each intervening DAR location INDEPENDENTLY. Relative to the 'anchor ;
* vector' that is formed by the MRD endpoints (origin and destination), ;
* 3 comparisons are made. Call the MRD origin A, and the destination B. ;
* Consider a temporally intervening DAR location, and call it X. ;
* Then construct 2 vector segments: A->X and X->B. ;
* ;
* Test 1: Direction ;
* Is the azimuth of A->X is within +/- the user-defined threshold the X ;
* is flagged for retention. ;
* ;
* Test 2: Angle ;
* If the angle formed by A->X->B is greater than a user-defined threshold;
* then X is flagged for retention. Note that if X were directly along ;
* the A->B vector, the angle would be 180. ;
* ;
* Test 3: Percent Distance ;
* If the total distance of travel along A->X->B is no more than a user- ;
* defined threshold of the anchor-distance A->B, then X is flagged for ;
* retention. That is, how much farther (in %) are you willing to let ;
* your animal wander in order to pass through location X? ;
* ;
* Remember that each intervening DAR location that falls between the MRD ;
* anchor points is evaluated independently of any others. ;
* ;
* If location X does not pass any of the 3 test above, it is filter from ;
* the output "both" datasets. ;
* ;
* At this time, the results of the tests are stored in a variable called ;
* "bothtest". In the future, I will develop SAS code that allows the ;
* user to decide which of the 3 tests must be passed in order to retain ;
* the DAR location (eg. any one of the tests as the program is currently ;
* designed, or perhaps any two of the tests, or all 3 tests must pass, ;
* etc.) ;
* ;
* Before the "both" dataset is finalized, the preliminary results are ;
* passed through a RATE-ONLY filter to validate that no location is ;
* being retained that produces an implausible rate of movement. ;
* ;
* *****;
* *****;
* March 31, 2003 ;
* ;
* When removing redundant records, the best LC with > NOPC and > messages is ;
* sorted to the first (retained) record. ;
* ;
* proc sort data=apdata ;
* by line descending xrank descending iqx descending nmess ;
* ;
* NLOCRAK changed to: nlocrank = mrank*10 + iqx*10 + iqy + nmess/100 ;
* ;
* *****;

```

```

* *****;
* April 04, 2003 ;
* ;
* The user now specifies how many of the 3 directionality tests must pass in ;
* order for a DAR location to be amended with the MRD results in the 'Best ;
* hybrid' output file. Variable=TESTPASS. ;
* ;
* The DAR filter logic was changed significantly. ;
* If the angle is too acute, the location is flagged implausible. ;
* If the angle is OK, but RATE1 is GT MaxRate, the location is implausible. ;
* If the angle is OK, and RATE1 is OK, but RATE2 is GT MaxRate, then a distance ;
* test is implemented to decide if the first Vector-2 node is implausible, or ;
* if the second Vector-2 node should be retested in the next iteration. ;
* The test considers 4 locations: A, B, C, D. Location B is in question. ;
* If A->B->D is longer than A->C->D, then B is flagged implausible. But if not ;
* longer, then B is flagged OK, and it is assumed that location C (the 2nd ;
* Vector-2 node) will be removed during the next iteration because the excessive ;
* rate will be assigned to RATE1 when location C becomes Location B. ;
* The procedure is iterated 4 times. Then a 5th iteration is executed that uses ;
* the original 3-location (A,B,C) logic. [But Changed in Oct 2004 V64, it is ;
* iterated 3 times and the original 3-loc logic is not executed.] ;
* ;
* Similarly, when the MRD locations and the 'Best Hybrid' locations are filtered ;
* for RATE-only (last step), the logic uses the 4-location method for 2 iterations; ;
* and then the original 3-location method for a final 3rd iteration. ;
* [But Changed in Oct 2004 V64, it is iterated 3 times and the original 3-loc ;
* is not executed.] ;
* ;
* REMOVED the options for RATE and ANGLE, and for ANGLE ONLY ;
* *****;
* *****;
* May 09, 2003 v61 ;
* ;
* Implemented the use of calc_dar2.sas which calculates distances using a ;
* dynamic estimate of the earth's radius in the vicinity of the argos locations. ;
* Previous versions used a fixed earth radius, which would introduce minor ;
* inaccuracy in the distance calculation between 2 argos locations, since the ;
* earth is not a perfect sphere. ;
* *****;
* *****;
* Sept 16, 2003 v62 ;
* ;
* Relaxed the requirement for detecting DIAG data redundancy. It was apparent ;
* that many DIAG records were exactly the same, except that the date/time ;
* differed by one second. Now, if ptt=ptt and primary lat/longs are equivalent ;
* and the date/times are within 2 seconds, the pass is declared redundant and ;
* the seconds variant is removed. ;
* *****;
* *****;
* October 01-15, 2004 v63 then v64 ;
* ;
* Modified DIAG input statements to accomodate a compressed ADS format that had ;
* not been documented until September 8, 2004. Francis Wiese first noted the ;
* problem in V62. The new format had blank fields in columns 1-2, rather than ;
* just column 1. Argos switched to a Linux ADS server in September. ;
* ;
* Also changed the input statements for ingesting pass DUR and NOPC. They ;
* will be retained into the final output datasets correctly. ;
* ;
* Wrap-up DAR (rate only) checks were interfering with retainment of original MRD ;
* anchor points. This error is now resolved so original MRD validations are not ;
* changed when the BOTH datasets are merged, EXCEPT when a PAIR of MRD locations ;
* has excessive RATES for vectors A->B and C->D, where A,B,C,D are 4 consecutive ;
* locations, and B-C are the MRD pair. This modification effectively filters the ;
* original MRD results for the "unlikely" condition that an erroneous "pair" of ;
* of MRD locations can be detected by implausible RATES for both the outbound and ;
* inbound vectors that precede and follow the suspect "pair". ;
* ;
* The final wrap-up DAR (rate only) check for the BOTH dataset, now uses a ;
* simplified RATE and MRD check. If the location in question has a ratel or rate2;
* that exceeds the rate threshold, and dist1 and dist2 are both GT maxredun, the ;

```

```

* location is filtered. In other words, the complex steps below were removed: ;
* If A->B->D is longer than A->C->D, then B is flagged implausible. But if not ;
* longer, then B is flagged OK, and it is assumed that location C (the 2nd ;
* Vector-2 node) will be removed during the next iteration because the excessive ;
* rate will be assigned to RATE1 when location C becomes Location B. ;
* ;
* ***** ;
* ***** ;
* January 10, 2005 v65 ;
* ;
* Included a new user defined option to select the equation for ranking location ;
* quality within duty cycle. The ranking decides which location within a duty ;
* cycle will be selected for the "pick of the day" datasets. ;
* ;
* rankmeth 1: nlocrank = mrank*10 + iqx*10 + iqy + nmess/100 ;
* rankmeth 2: nlocrank = mrank*100 + iqx*100 + nmess + iqy/10 ;
* rankmeth 3: nlocrank = mrank*1 + nmess ;
* ;
* The first method (#1) is the same implemented for Version 6.0-6.4 ;
* The second method (#2) is like #1 except it prioritizes nmess over iqy. ;
* The third method (#3) is the same implemented for Versions 5 and earlier. ;
* ;
* Another change is the user-option ability to specify different TESTPASS ;
* thresholds for location classes Z&B, and LCs A&0. ;
* ;
* For example, ;
* %let testp_0a = 2 ;
* %let testp_bz = 3 ;
* ;
* ***** ;
* ***** ;
* November 16, 2005 v66 ;
* ;
* Earlier code was added to reimplement the Weeding locations based on MRD ;
* It was conducted in the event that a location was removed by the RATE-checking ;
* that was previously defining redundancy. ;
* ;
* In November 2005, the 24 hr/day duty cycle of the walrus ptt's failed in this ;
* section of code. It was determined that the count variable needed to be dropped ;
* Version 6.6 drops the count variable, which fixed 24 h/d walrus data ;
* ;
* data weeddist(drop=weeddist count) le2 gis(drop=weeddist count) ;
* set gis ;
* ***** ;
* ***** ;
* January 14, 2006 v67 ;
* ;
* Included a section of code in the DIAG ingestion to read the new compressed ;
* Argos DIAG format which had removed the blank space in column 1. ;
* This was done at the request of Kirk Bates, Idaho Raptor Researcher ;
* ;
* ***** ;
* ***** ;
* March 29, 2006 v68 ;
* ;
* Included two new output variables: ;
* dist2dply - Great-circle distance between the location and the deployment loc. ;
* azfrmdply - Great-circle bearing from the deployment loc to the Argos loc. ;
* ;
* These variables are useful for quantifying location accuracy for ptt's at a known ;
* stationary location. Define the known location as the "deployment location", ;
* and define a unique deployment-logic-block to temporally coincide with the ;
* period when the ptt remained at the known location. ;
* ***** ;
* ***** ;
* April 3, 2006 v70 ;
* ;
* Animal can now be 25 characters in length, but NO blank spaces (Fixed in 7.02). ;
* ;
* Changed User Deployment Blocks to External ASCII file. ;

```

