U.S. DEPARTMENT OF TRANSPORTATION
FEDERAL AVIATION ADMINISTRATION

National Policy

ORDER 8260.58A

Effective Date:
03/14/2016

SUBJ: United States Standard for Performance Based Navigation (PBN) Instrument Procedure Design

This order prescribes standardized methods for designing and evaluating Performance Based Navigation (PBN) instrument flight procedures (IFPs) in the United States and its territories.

Performance Based Navigation (PBN) describes an aircraft's capability to navigate using performance standards and is comprised of Area Navigation (RNAV) and Required Navigation Performance (RNP). RNAV enables aircraft to fly on any desired flight path within the coverage of ground or space based navigation aids, the limits of the capability of the self-contained systems, or a combination of both capabilities. RNP is RNAV with the addition of onboard performance monitoring and alerting capability. A defining characteristic of RNP operations is the ability of the aircraft navigation system to monitor the navigation performance it achieves and inform the crew if the requirement is not met during an operation. The performance requirements of PBN are conveyed to the users through navigation specifications.

This order is to be used by all personnel responsible for the preparation, approval, and promulgation of PBN IFPs.

John S. Duncan
Director, Flight Standards Service

Distribution: Electronic Only Initiated By: AFS-400

THIS PAGE IS INTENTIONALLY LEFT BLANK

Table of Contents

Chapter 1. General Information

Section 1-1. Purpose

1-1-1. Purpose of this order. This order provides guidance for the design and evaluation of Performance Based Navigation (PBN) Instrument Flight Procedures (IFPs). For the purposes of this order, PBN IFPs are those based on Area Navigation (RNAV) or Required Navigation Performance (RNP) to include transitions to an Instrument Landing System (ILS) or Ground Based Augmentation (GBAS) Landing System (GLS) final segment [see appendix C and Order 8260.19, section 4-6 and chapter 8 paragraph 'Equipment requirements notes'.

1-1-2. Audience. Al personnel who are responsible for IFP development and/or evaluation.

1-1-3. Where you can find this order. You can find this order on the Federal Aviation Administration's (FAA) Web site.

1-1-4. What this order cancels.

a. Order 8260.53 titled "Standard Instrument Departures that Use RADAR Vectors to Join RNAV Routes" dated 07/25/2005.

b. Order 8260.58 titled "United States Standard for Performance Based Navigation (PBN) Instrument Procedure Design" dated 09/21/2012.

Section 1-2. Basic Criteria

1-2-1. General.

a. Source directive. FAA Order 8260.3, United States Standard for Terminal Instrument Procedures (TERPS) applies unless otherwise specified by this directive.

b. Related directives. The following orders are referenced in the directive or otherwise related to the development/documentation of PBN procedures.

(1) Order 8260.19, Flight Procedures and Airspace.

(2) Order 8260.42, United States Standard for Helicopter Area Navigation (RNAV).

(3) Order 8260.46, Departure Procedures (DP) Program.

c. Scope. This chapter contains only that information common to all PBN IFPs. Criteria which do not have general application are located in the chapter's specific to the flight phase, navigation type, or application.

1-2-2. Formulas/Calculations. Formulas in this directive use degree calculations. The examples provided use only two decimal places; therefore, the answers are approximate values. Actual calculations must use the maximum number of values after the decimal place for each step. Rounding of the final result should be for documentation purposes only. For the purposes of this order, the term "r" represents the mean radius of the earth used in spherical calculations and is equal to 20890537 feet. For mathematics convention and other common equation terms see Order 8260.3, appendix D.

1-2-3. Geospatial standards. PBN IFPs must be evaluated geodetically [see appendix E].

1-2-4. PBN concept. The PBN concept is made up of three interrelated elements: the navigation specification, the NAVAID infrastructure, and the navigation application. These three elements define the minimum required aircraft/system performance used as the basis for the obstacle evaluation areas described in this order.

a. Navigation specification (NavSpec). The NavSpec is a set of aircraft/aircrew certification and operational approval requirements that support a navigation application within a defined airspace concept. NavSpecs which require on-board performance monitoring and alerting are termed RNP specifications. Those that do not require on-board performance monitoring and alerting are termed RNAV specifications. The NavSpec defines the performance required by an RNAV or RNP system as well as any functional requirements (e.g., ability to conduct curved path legs). See AC 90-101 and AC 90-105 for definitions. NavSpecs include:

(1) Performance. The performance required in terms of accuracy, integrity, and continuity.

(2) Functions. The functions available to achieve the required performance.

(3) Navigation sensors. The navigation sensors that may be used to achieve the required performance.

(4) Procedures. Flight crew and other procedures to achieve the required performance.

b. NAVAID infrastructure. NAVAID infrastructure relates to space or ground-based NAVAIDS addressed in the NavSpec.

c. Navigation application. Also referred to as flight phase. A navigation application is the operation in which a specific NavSpec and associated performance requirements applies (e.g., ATS route, IFP segment, etc.).

Table 1-2-1. Navigation Accuracy by NavSpec/Flight Phase

Navigation specification	Flight phase						
	En route	STAR/ Feeder/TAA	Initial	Intermediate	Final	Missed[1]	Departure
RNAV 2	2	2					2
RNAV 1[2]		1	1	1		1	1
RNP 2	2						
RNP 1[2]		1	1	1		1	1
RNP APCH[3]			1	1	0.3[4]/40m[5]	1	
A-RNP[6]	2 or 1	1 – 0.3	1 -0.3	1- 0.3		1- 0.3	1
RNP AR APCH			1 to 0.1	1 to 0.1	0.3 to 0.1	1 to 0.1	
RNP 0.3[7]	0.3	0.3	0.3	0.3		0.3	0.3

[1] Missed approach section 2 only.

[2] STAR/Feeder/TAA, departure, initial, intermediate, and missed section 2. Beyond 30 NM from ARP the effective XTT for the purpose of IFP design is 2.

[3] RNP APCH part A is enabled by GNSS and baro-VNAV and part B is enabled by SBAS.

[4] Part A only.

[5] Part B along-track performance is 40 meters; angular performance requirements apply to lateral.

[6] A-RNP permits a range of scalable RNP lateral navigation accuracies.

[7] Primarily intended for helicopter operations only. RNP APCH applies to the final flight phase.

1-2-5. Obstacle Evaluation Area (OEA) construction.

a. Course.

(1) Lateral and longitudinal dimensions. Construct straight-line courses as a geodesic path. Construct parallel and trapezoidal boundary lines as a locus of points measured perpendicular to the geodesic path. Determine OEA lateral boundary relative to course centerline using ellipsoidal calculations.

(2) Vertical dimension (surface elevations). Obstacle clearance evaluation/clearance surfaces are generally defined as either a level or sloping plane relative to a defined origin MSL elevation. Evaluate the orthometric height above the geoid (MSL) of obstacles and terrain relative to these surfaces.

(3) Alignment. The alignment tolerance is ± 0.03 degrees of all course differentials. The general alignment restriction between TF/CF legs at or above FL 195 is limited to a maximum course change of 70 degrees, while below FL 195 is 90 degrees; however, specific

requirements stated in a segment’s chapter/section take precedence. When preceded by a DF leg, the alignment restriction applies to both the early and late tracks. The general alignment restriction between TF and RF legs is no course change exceeding alignment.

Note: When using pre-established fixes, acceptably small differences in fix coordinates can cause legs to exceed alignment tolerance. The alignment tolerance can be exceeded when the calculated ideal fix coordinates round to match the pre-existing fix coordinates [see Order 8260.19, paragraph 8-5-2].

b. Area dimensions. For IFP design purposes, the term XTT is the larger of basic XTT and the effective XTT associated with a leg.

(1) Length.

(a) Minimum length (fix-to-fix). Generally, minimum leg length is the lesser of 2 × XTT or 1 NM, but where applicable may also be no less than;

1. The sum of the distance of turn anticipation (DTA) for each fly-by (FB) turn [see formula 1-2-1].

Note: Not applicable for FB turns of 10 degrees or less.

Formula 1-2-1. Distance Turn Anticipation

$$\mathrm{DTA} = \mathrm{R} \times \tan\left(\frac{\beta}{2}\right)$$

Where:
β = magnitude of heading change in degrees
R = turn radius

Example:

$$\mathrm{DTA} = 3.52 \times \tan\left(\frac{75}{2}\right)$$
$$\mathrm{DTA} \approx 2.7$$

Note: This formula produces a value with the same unit as R.

2. For TF legs following a fly-over (FO) turn, a length sufficient to allow the aircraft to return to course centerline. Determine using formula 1-2-2.

Formula 1-2-2. TF Minimum Leg Length Following FO Turn

If $\beta_i < \text{acos}(\sqrt{3} - 1)$ use:

$$\text{L=R}_\text{i} \times \left[\sin(\beta_\text{i}) + 2 \times \sin\left(\text{acos}\left[\frac{1 + \cos(\beta_\text{i})}{2}\right]\right)\right] + \text{R}_\text{ii} \times \tan\left(\frac{\beta_\text{ii}}{2}\right)$$

If $\beta_\text{i} \geq \text{acos}(\sqrt{3} - 1)$ use:

$$\text{L} = \text{R}_\text{i} \times [\sin(\beta_\text{i}) + 4 - \sqrt{3} - \sqrt{3} \times \cos(\beta_\text{i})] + \text{R}_\text{ii} \times \tan\left(\frac{\beta_\text{ii}}{2}\right)$$

Note: L must not be less than 1 NM

Where:
R_i = turn radius at the first fix
β_i = magnitude of heading change at the first fix
R_ii = turn radius at the second fix
β_ii = magnitude of heading change at the second fix

Note: When multiple courses precede the FO fix (e.g., VA-DF legs) use the track resulting in the greatest magnitude of heading change at the first fix.

Example:

$$\text{L} = 5.6 \times \left[\sin(33) + 2 \times \sin\left(\text{acos}\left[\frac{1 + \cos(33)}{2}\right]\right)\right] + 6.2 \times \tan\left(\frac{45}{2}\right)$$

$$\text{L} \approx 10.03$$

Note: This formula produces a value with the same unit as R.

(b) RF leg length may be calculated using formula 1-2-3. Use in conjunction with formula 1-2-4 when necessary to calculate an RF leg of a specific length.

Formula 1-2-3. RF Leg Length

$$\text{L} = \frac{\alpha \times \pi \times \text{R}}{180}$$

Where:
α = degrees of arc
R = arc radius

Example:

$$\text{L} = \frac{98.9 \times \pi \times 4.2}{180}$$

$$\text{L} \approx 7.25$$

Note: This formula produces a value with the same unit as R.

Formula 1-2-4. Degrees of an Arc

$$\alpha = \frac{180 \times \mathrm{L}}{\pi \times \mathrm{R}}$$

Where:
L = arc length
R = arc radius

Note: This formula works with any unit so long as L & R share the same unit.

Example:

$$\alpha = \frac{180 \times 7.25}{\pi \times 4.2}$$

$$\alpha \approx 98.90^\circ$$

(c) Maximum length (fix-to-fix). Maximum Standard Instrument Approach Procedure (SIAP) segment length is as specified in Order 8260.3, chapter 2. In no case should the length of any PBN leg exceed 500 NM.

(d) Along-track tolerance (ATT). OEAs are constructed and evaluated from ATT prior to a leg's initial fix (early) to ATT past its termination fix (late). ATT is equal to the navigation accuracy associated with the flight phase [see table 1-2-1]. ATT (same or adjacent legs) can overlap when legs meet minimum length requirements.

1. Except where specified otherwise (e.g., intermediate segment) use the ATT specific for the leg being constructed for both early and late ATT.

2. The baseline used for determining early/late ATT area depends on the leg/turn type**.**

a TF (no turn) and FO turn. ATT is relative to a line perpendicular to the inbound leg at the fix [see figure 1-2-1].

b FB turn. ATT is relative to the bisector used for OEA construction [see figure 1-2-2].

c RF. ATT is relative to a line drawn radially from the turn center point. Distance is measured from the fix along the RF arc [see figure 1-2-3].

Figure 1-2-1. ATT Straight

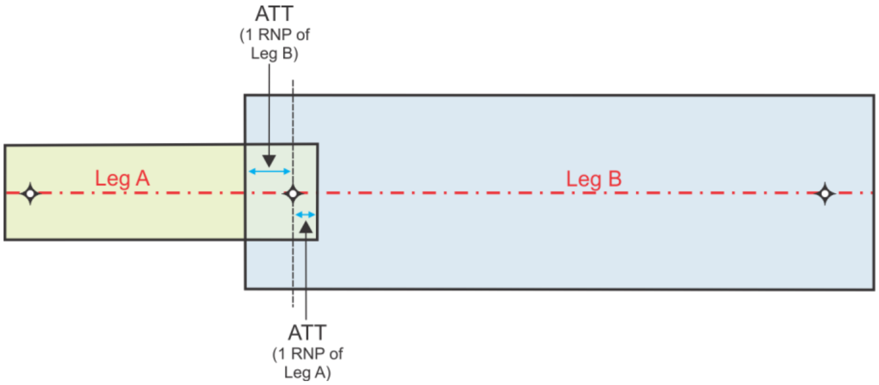


Figure 1-2-2. ATT Turning

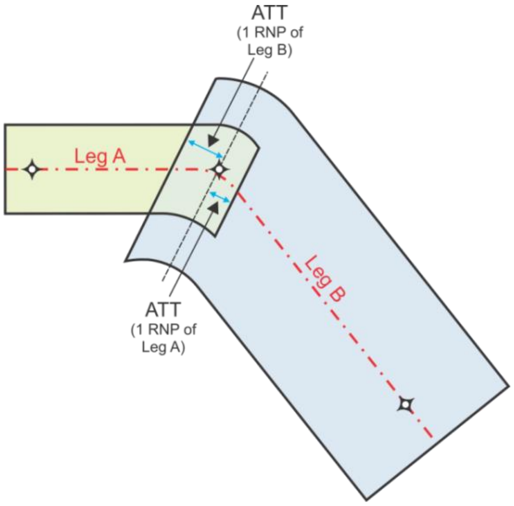
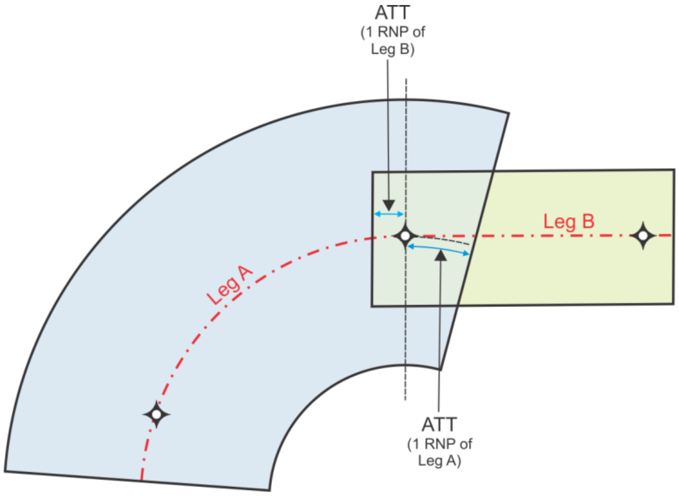


Figure 1-2-3. ATT RF



(2) Width.

(a) Cross-track tolerance (XTT). XTT is the navigation accuracy requirement specified in table 1-2-1 based on NavSpec and flight phase. XTT is used to determine OEA lateral dimensions. Primary area width is 2 × XTT either side of course and secondary area width is 1 × XTT either side of the primary area, if applicable. The XTT for A-RNP and RNP AR is the specified RNP value for the leg being evaluated. Reference to route width is often specified as NM values measured from secondary area edge across the primary area to the secondary edge at the other side. For example, feeder widths may be referred to as “2-4-4-2.”

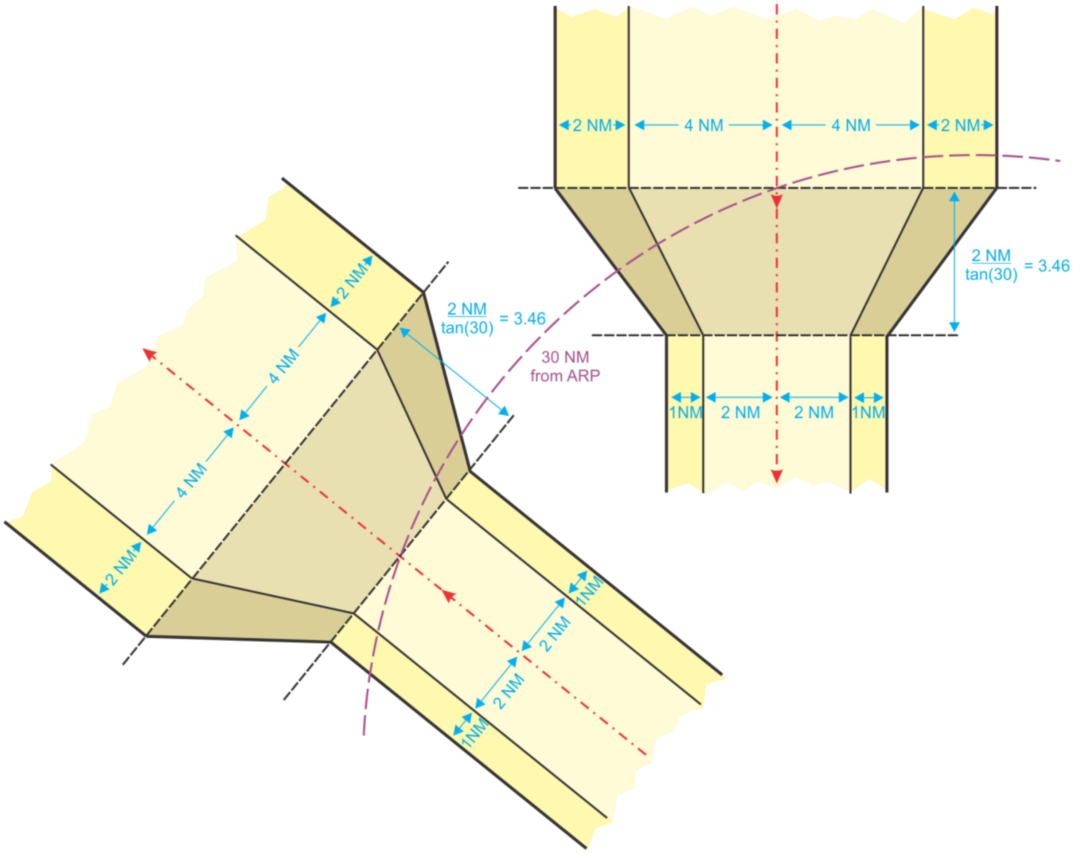
(b) Width changes. RNP AR APCH width changes occur abruptly at the start fix of a leg with a new RNP. A-RNP width changes are TBD. For all other IFPs, OEA construction accommodates width changes at the transition to the final approach mode and at the transition to or from the en route mode. See the applicable chapter for width changes at the final approach transition. For the en route transition, legs that cross a point 30 NM from ARP, the OEA primary area width expands or tapers at a rate of 30 degrees relative to course to the appropriate width [see figure 1-2-4 and figure 1-2-5]. Calculate the total along-track distance required to complete the transition using formula 1-2-5. STAR/Feeder and approach legs designed to cross within 30 NM of the ARP more than once do not change to RNAV/RNP 1 (terminal mode) until the 30 NM limit is crossed for the last time. A departure or missed approach segment designed to cross a point 30 NM from the ARP more than once changes to RNAV/RNP 2 (en route mode) and remains expanded when it crosses the boundary the first time.

1. Non-RF width changes.

a Inbound. The secondary area tapers from en route mode width when the 30 NM point is crossed to terminal mode width abeam the point the primary area is fully tapered. Leg width tapers regardless of fix location within the tapering section unless a turn is associated with the fix. Delay OEA taper until the turn is complete and normal OEA turn construction is possible.

b Outbound. The secondary area expands from terminal mode width when the 30 NM point is crossed to en route mode width abeam the point the primary area is fully expanded. Leg width expands regardless of fix location within the expansion section unless a turn is associated with the fix. Start the OEA expansion so as to achieve full expansion to en route mode width prior to the turn.

Figure 1-2-4. Non-RF Width Change at 30 NM from ARP



2. RF width changes. Calculate the perpendicular distance from the RF leg track centerline to primary and secondary boundaries at any along-track distance (specified as degrees of RF arc) from the point the track crosses the 30 NM point using formula 1-2-6 [see figure 1-2-5]. The width transition must be fully contained within a single RF leg.

Formula 1-2-5. RF Width Change Distance at 30 NM from ARP

$$D = \frac{2 \ \times \mathrm{XTT}_{\mathrm{large}} - 2 \ \times \mathrm{XTT}_{\mathrm{small}}}{\tan(30)}$$

Where:
$\mathrm{XTT}_{\mathrm{large}}$ = the largest XTT value
$\mathrm{XTT}_{\mathrm{small}}$ = the smallest XTT value

Note: This formula works with any unit so long as all XTT values share the same unit.

Example:

$$D = \frac{2 \times 2 - 2 \times 1}{\tan(30)}$$

$$D \approx 3.46$$

Formula 1-2-6. RF Leg XTT Change

$$\text{Boundary}={}^{1}/_{2}\,\text{Width}_{\text{large}}-\frac{{}^{1}/_{2}\text{Width}_{\text{small}}\times \text{X}_{\text{angle}}\times \text{R}\times\pi}{\text{D}\times 180}$$

Where:

${}^{1}/_{2}\,\text{Width}_{\text{large}}$ = 2 × largest XTT for primary area or 3 × largest XTT for secondary area

${}^{1}/_{2}\,\text{Width}_{\text{small}}$ = 2 × smallest XTT for secondary or 3 × smallest XTT for secondary area

X_{angle} = angle relative to center point from beginning of taper to point of interest

R = arc radius

D = along track distance required to complete the XTT change

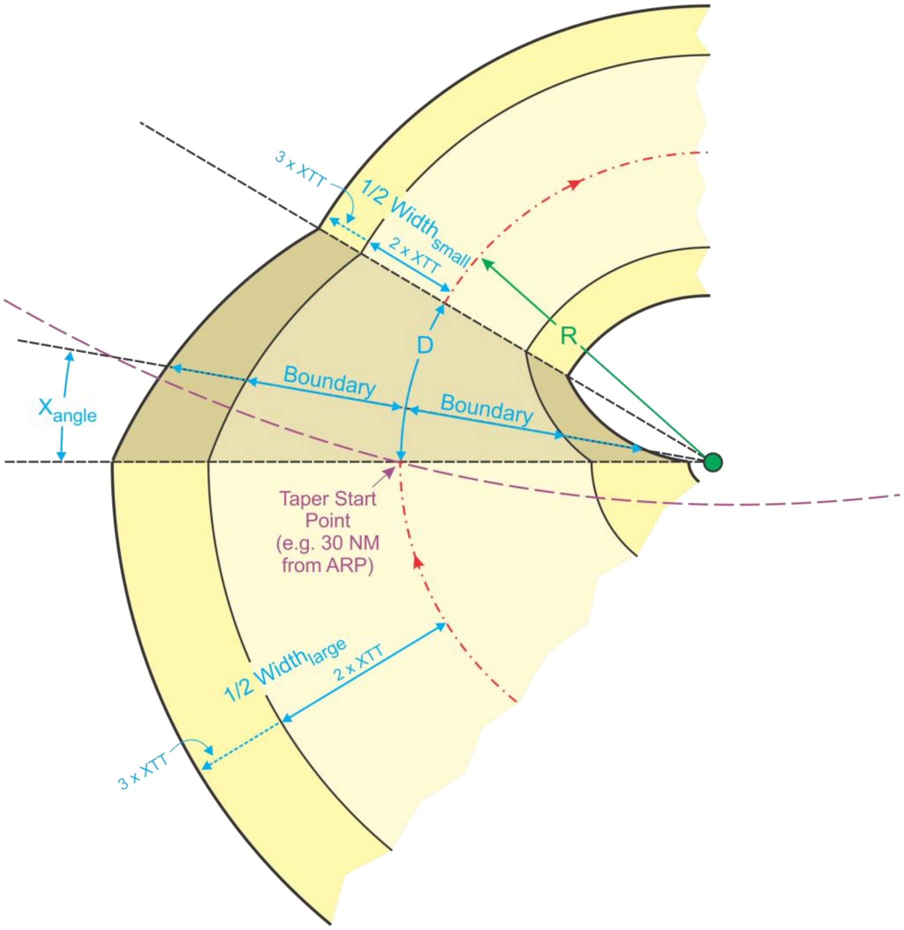
Note: This formula works with any unit so long as all XTT values share the same unit.

Example:

$$\text{Boundary}=6-\frac{3\times 10\times 6.44\times\pi}{3.46\times 180}$$

$$\text{Boundary}\approx 5.03$$

Figure 1-2-5. RF Width Change at 30 NM from ARP



c. Turn parameters. For OEA construction, a turn is indicated when the course change exceeds the alignment tolerance of 0.03 degrees [see paragraph 1-2-5.a(3)].

(1) Altitude. Calculate the assumed altitude at a turn fix/point as specified below. For other cases [e.g., Q or T routes, or standard terminal arrival (STAR) or standard instrument departure (SID) legs that precede/follow manual termination legs that make it impractical to determine a starting altitude/point] use the highest anticipated altitude at the turn fix based on specified altitudes, airspace, and anticipated use.

Note: Use of the highest anticipated altitude is required even when obstacle clearance is assured for flyability, path repeatability, and airspace/route separation.

(a) Calculate projected altitude. Except for turns inside the Precision Final Approach Fix (PFAF), projected altitudes are calculated using formula 1-3-8 based on the rise of an assumed vertical path from a specified start point.

1. Final approach segment.

a Turns inside the PFAF. Use the final segment vertical path, i.e., glidepath angle (GPA). Determine the projected altitude by calculating the vertical rise from the Landing Threshold Point (LTP)/Fictitious Threshold Point (FTP) using formula 1-3-4.

b Turns at the PFAF. There is no projected altitude. Use the specified PFAF altitude.

2. Intermediate and initial approach segments. The vertical path is 250 ft/NM (CAT A-E) or 400 ft/NM (COPTER). Determine the projected altitude by calculating the vertical path rise fix-to-fix from the start point/altitude to the turn fix. For the first leg, the start point is the PFAF at the specified PFAF altitude. For subsequent legs, start at the end fix of the preceding leg at the assumed fix altitude.

3. Feeder segments and STAR runway transitions that terminate on an instrument approach procedure (IAP). The vertical path is 250 ft/NM (CAT A-E) (COPTER TBD). Determine the projected altitude by calculating the vertical path rise fix-to-fix from the start point/altitude to the turn fix.

a Feeder. For the first leg, start at the initial approach fix (IAF) at the assumed IAF altitude. The start point/altitude for subsequent legs is the end fix at the assumed fix altitude of the preceding leg.

b STARs. For the first leg, start at the IAP fix at the assumed fix altitude. For subsequent legs, start at the end fix preceding leg at the assumed fix altitude. If a STAR transition serves more than one IAP, base the STAR on the IAP with the highest assumed altitude.

4. Missed approach (MA) and departure (prior to transition route). For MA legs, the vertical path is the higher of 250 ft/NM (CAT A-B), 500 ft/NM (CAT C-E), 400 ft/NM (COPTER), or the specified climb gradient (CG) in effect at the point of interest. For departure

legs (all CATs), the vertical path is the higher of 500 ft/NM below 10000 MSL and/or 350 ft/NM 10000 MSL and above (CAT A-E), 400 ft/NM (COPTER) or the specified CG in effect at the point of interest. For MA, the start point for the first leg is the SOC applicable to the approach type [see applicable chapter]. For departures, the start point for the first leg is the departure end of runway (DER) at DER elevation.

a Turn-at-altitude. To calculate the highest altitude in the turn used for OEA construction, see paragraph 1-3-1.g(3); otherwise use the specified turn altitude. The start point for subsequent legs is the point along the course centerline where the specified turn altitude is reached. Calculate the vertical rise from the start point measured direct to subsequent fix.

b All other turns. Calculate the projected fix altitude by applying the vertical path rise from the start point/altitude along the course centerline to the subsequent fix/point. For fixes following a DF leg, the course centerline is a direct line from the turn point/fix to the subsequent fix. The start point for subsequent legs is the end fix/point of the preceding leg at the assumed fix/point altitude.

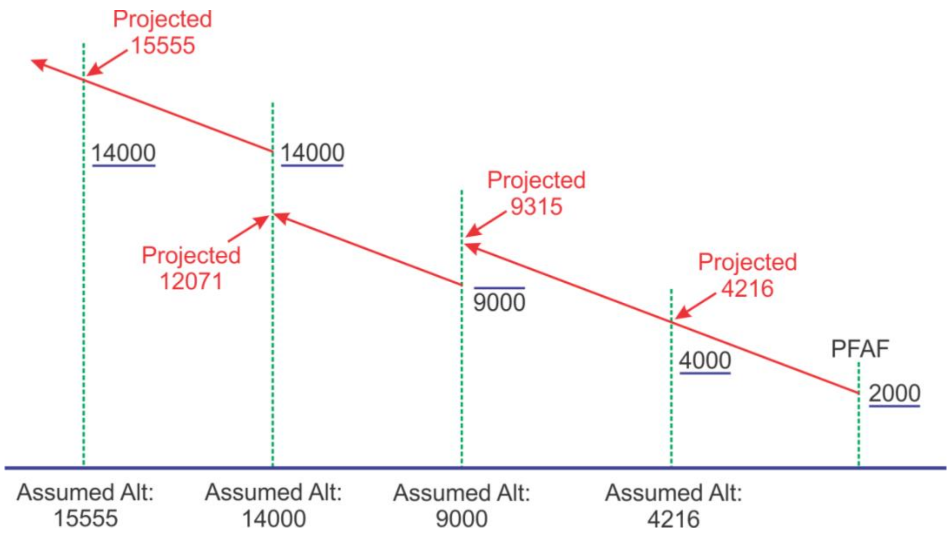
(b) Compare to published altitudes. Compare projected altitude(s) to published altitude(s) to include minimum en route altitudes (MEAs).

(c) Determine assumed altitude [see figure 1-2-6]. The assumed altitude used for turn construction is:

1. The higher of the projected altitude and any published minimum or mandatory altitude.

2. The lower of the projected altitude and any maximum altitude, to include the overall maximum altitude applicable to the flight phase (e.g., missed approach clearance limit altitude). The maximum altitude at a fix or for the flight phase can also be specified by ATC.

Figure 1-2-6. Assumed Altitudes for OEA Turn Construction



(2) Airspeed. Locate and use the appropriate KIAS from table 1-2-2. Determine the KTAS for the turn using formula 1-2-7 based on the assumed altitude at the turn fix/point [see paragraph 1-2-5.c(1)].

Table 1-2-2. Indicated Airspeeds (KIAS)

Flight Phase		Indicated Airspeed by CAT				
		A	B	C	D	E
At or Above 10000 feet						
En route, STAR/Feeder/TAA, Initial, Intermediate, Missed, Departure		180	250	300	300	350
Below 10000						
En route, STAR/Feeder/TAA, Initial, Intermediate		150	180	250	250[1]	310
Final		90	120	140	165	250
Missed Approach (MA), Departure		110	150	240	265	310
Minimum Airspeed Restriction						
Minimum Airspeed Restriction[2]	STAR/Feeder/TAA, Initial, Departure	110	140	200[3]	210[3,4]	310
	Intermediate	110	140	180	180	310
	Missed Approach	100	130	165	185	310
	Final	Not Authorized				

[1] Consider using 265 KIAS where heavy aircraft routinely exceed 250 KIAS under 14 CFR § 91.117.

[2] Airspeed restrictions may be established at a charted fix to reduce turn radius, avoid obstacles, accommodate ATC request, etc. Use the fewest number of restrictions in the same IFP possible. Especially avoid consecutive restrictions requiring speed changes of less than 20 KIAS in the same or adjoining segments. Flight Standards or military authority approval is required for missed approach restrictions for other than obstacle avoidance.

[3] 250 above 10000 feet except for initial and/or STAR termination fix.

[4] 200 underlying Class B airspace per 14 CFR § 91.117(c).

Formula 1-2-7. True Airspeed

$$V_{\mathrm{KTAS}} = \frac{V_{\mathrm{KIAS}} \times 171233 \times \sqrt{303 - 0.00198 \times \mathrm{alt}}}{(288 - 0.00198 \times \mathrm{alt})^{2.628}}$$

Where:
V_{KIAS} = indicated airspeed for applicable category and segment combination

Example:

$$V_{\mathrm{KTAS}} = \frac{165 \times 171233 \times \sqrt{303\text{-}0.00198 \times 650}}{(288\text{-}0.00198 \times 650)^{2.628}}$$

$$V_{\mathrm{KTAS}} \approx 170.88$$

(3) Maximum bank angles. (Not applicable to COPTER procedures).

(a) Bank limited turns (assumed altitude at turn point/fix < 500 feet above airport elevation): 3 degrees.

(b) Low altitude transition turns (assumed altitude at turn point/fix from 500 feet above airport elevation up to and including FL 195):

1. XTT of the leg being constructed ≥ 1.0: 25 degrees

2. XTT of the leg being constructed < 1.0: 20 degrees

(c) High altitude transition turns (> FL 195): 5 degrees

(d) FB turns only: The lesser of one-half the track change (five degrees minimum) or the applicable low/high altitude transition bank angle. Where operationally beneficial (e.g., to reduce leg length) an exception to the one-half track change rule is authorized at a fix when the XTT for the outbound leg is not less than 1.0, a succeeding turn fix uses the applicable low/high altitude transition bank angle, and/or the succeeding leg meets minimum length requirements.

(4) Tailwind (V_{KTW}). Calculate the appropriate tailwind for the assumed altitude at the turn fix [see paragraph 1-2-5.c(1)] using formula 1-2-8. Where operationally advantageous the 99th percentile wind speed values determined from analysis of a five-year locally measured database may be substituted [see appendix A].

Formula 1-2-8. Assumed Tailwind

$$V_{KTW} = 0.00198 \times alt + 47$$

Note: If $apt_{elev} - alt \leq 2000$, $V_{KTW} = 30$

Example:

$V_{KTW} = 0.00198 \times 3400 + 47$
$V_{KTW} \approx 53.73$

(5) Turn radius (R). The design turn radius value is based on the turn parameters assumed tailwind [see formula 1-2-8], groundspeed [see formula 1-2-9], assumed altitude at the turn fix/point [see paragraph 1-2-5.c(1)], and bank angle [see paragraph 1-2-5.c(3)]. Calculate R using formula 1-2-10. For RF turn radius, see paragraph 1-2-5.d(3).

Formula 1-2-9. Groundspeed

If alt $>$ FL 195 use:

$$\mathrm{V_{ground}} = \text{lesser of } 570 \text{ or } \frac{0.9941 \times \mathrm{alt}}{100} + 287$$

If alt $\leq$ FL 195 use:

$$\mathrm{V_{ground}} = \text{lesser of } 500 \text{ or } \mathrm{V_{KTAS}} + \mathrm{V_{KTW}}$$

Where:
$\mathrm{V_{KTAS}}$ = calculated KTAS using alt and applicable KIAS
$\mathrm{V_{KTW}}$ = calculated or historical tailwind (knots) using alt

Example:

$$\mathrm{V_{ground}} = \frac{0.9941 \times 26500}{100} + 287$$
$$\mathrm{V_{ground}} \approx 550.44 \text{ knots}$$

Formula 1-2-10. Turn Radius

$$\mathrm{R} = \frac{\mathrm{V_{ground}}^2}{\tan(\phi) \times 68625.4}$$

For FB turns, if resulting DTA $>$ 20 NM, then:

$$\mathrm{R} = \frac{20}{\tan(\beta \times 0.5)}$$

Where:
β = magnitude of heading change in degrees
$\mathrm{V_{ground}}$ = ground speed (knots)
ϕ = designed bank angle or 5 degrees where $>$ FL 195

Example:

$$\mathrm{R} = \frac{550.44^2}{\tan(5) \times 68625.4}$$
$$\mathrm{R} \approx 50.46 \text{ NM}$$

Formula 1-2-11. RF Bank Angle

$$\phi = \operatorname{atan}\left(\frac{{V_{\text{ground}}}^{2}}{R \times 68625.4}\right)$$

Where:
V_{ground} = calculated ground speed (knots) at the highest assumed altitude in the leg
R= turn radius (NM)

Example:

$$\phi = \operatorname{atan}\left(\frac{550.44^{2}}{50.46 \times 68625.4}\right)$$

$$\phi \approx 5.00°$$

d. Turn construction.

(1) Turns at FB fixes [see figure 1-2-7].

(a) *Step 1*. Establish a line through the turn fix that bisects the turn angle. With the origin on the bisector line, scribe an arc that is tangent to the inbound and outbound centerlines using a radius equal to R.

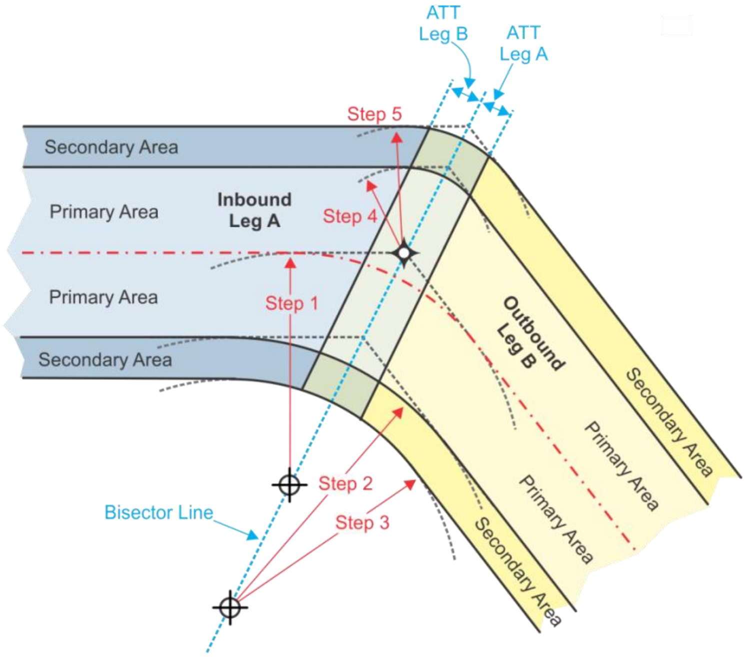
(b) *Step 2*. Primary area inner boundary. With the origin on the bisector line, scribe an arc that is tangent to the inner primary boundaries of the two legs using a radius equal to $R + XTT$.

(c) *Step 3*. Secondary area inner boundary (if applicable). Using the origin from *Step 2*, scribe an arc that is tangent to the inner secondary boundaries of the two legs using a radius equal to R.

(d) *Step 4*. Primary area outer boundary. Centered on the turn fix, scribe the primary area outer boundary arc using a radius equal to $2 \times XTT$.

(e) *Step 5*. Secondary area outer boundary (if applicable). Centered on the turn fix, scribe the secondary area outer boundary arc using a radius equal to $3 \times XTT$.

Figure 1-2-7. FB Turn Construction



(2) Turns at FO fixes [see figure 1-2-8 and figure 1-2-9].

(a) *Step 1.* FO turn construction incorporates a delay in start of turn to account for pilot reaction time and roll-in time. Calculate the extension distance using formula 1-2-12.

Formula 1-2-12. Reaction & Roll Distance

$$D_{rr}=\frac{V_{KTAS}\times 6}{3600}$$

Where:
V_{KTAS} = calculated KTAS using applicable altitude and KIAS combination

Example:

$$D_{rr}=\frac{253.62\times 6}{3600}$$

$$D_{rr}\approx 0.42\text{ NM}$$

(b) *Step 2.* Establish the baseline for construction of the turn expansion area as the line perpendicular to the inbound track at a distance past the turn fix equal to ATT + D_{rr}.

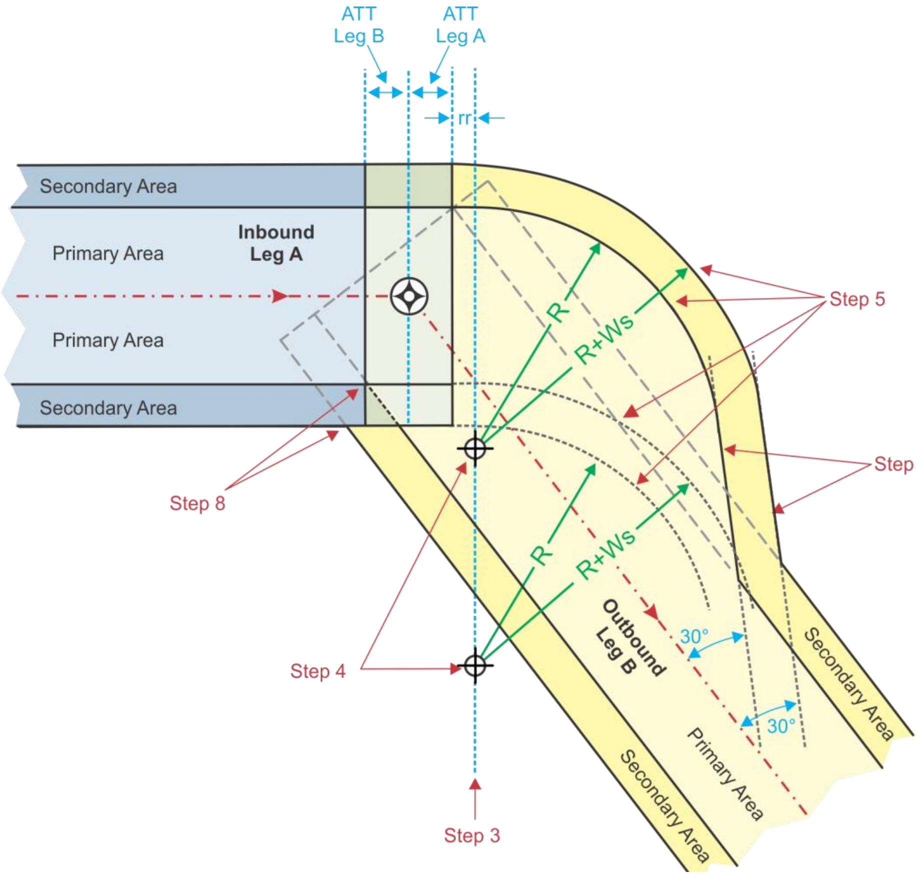
(c) *Step 3.* Determine R based on bank angle intended for design [see paragraph 1-2-5.c(3)].

(d) *Step 4*. On the baseline, locate the center points for the primary and secondary turn boundaries. The first is located at a distance R from the non-turning side primary boundary. The second is located at a distance R from the turning side secondary boundary.

(e) *Step 5*. From these center points construct arcs for the primary boundary of radius R. Complete the secondary boundary by constructing additional arcs of radius R + secondary area width from the same center points.

(f) *Step 6*. The arcs constructed in *Step 5* are tangent to the outer boundary lines of the inbound leg. Construct lines tangent to the arcs based on the first turn point tapering inward at an angle of 30 degrees relative to the outbound track that joins the arc primary and secondary boundaries with the outbound leg primary and secondary boundaries. If the arcs from the second turn point are inside the tapering lines as shown in figure 1-2-8, then they are disregarded and the expanded area construction is completed. If not, proceed to *Step 7*.

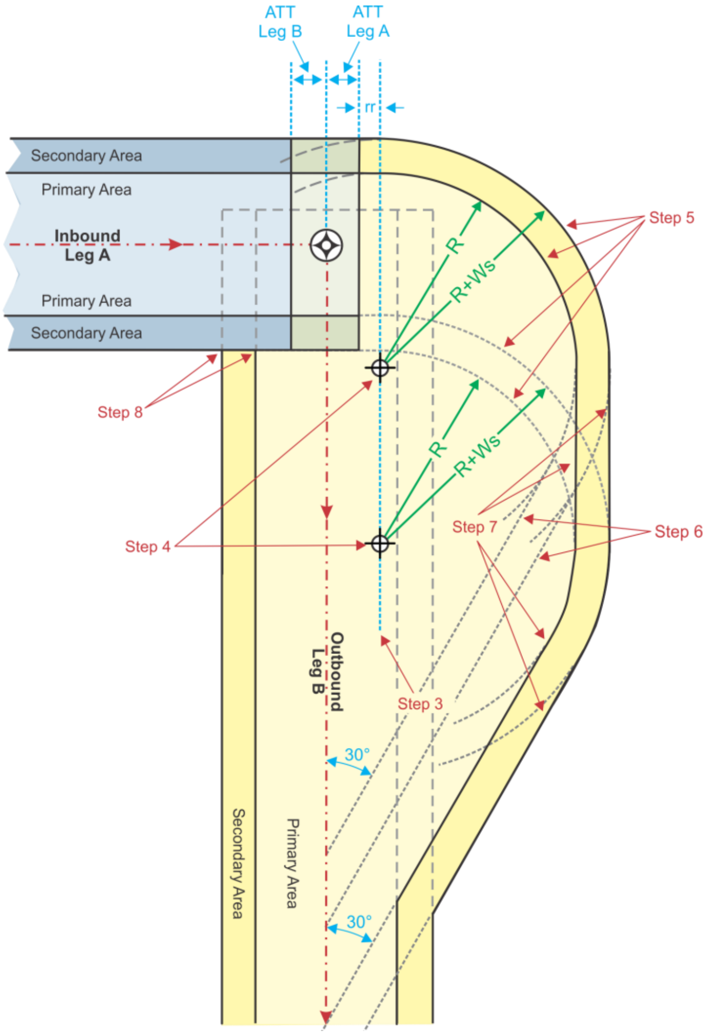
Figure 1-2-8. FO with No Second Arc Expansion



(g) *Step 7*. If both the inner and outer arcs lie outside the tapering lines constructed in *Step 6*, connect the respective inner and outer arcs with tangent lines and then construct the tapering lines from the arcs centered on the second center point.

(h) *Step 8*. The inside turn boundaries are the intersection of the preceding and succeeding leg primary and secondary boundaries.

Figure 1-2-9. FO with Second Arc Expansion



(3) Radius-to-fix (RF) Turn [see figure 1-2-10]. RF legs are used to control the ground track of a turn where obstructions prevent the design of a FB or FO turn, or to accommodate other operational requirements. Incorporation of an RF leg limits IFP availability to some users; therefore use of an RF leg requires Flight Standards approval except for procedures/segments based on the RNP AR APCH. OEA construction limits turn radius to a minimum value equal to or greater than the OEA (primary and secondary if applicable) half-width. The RF leg OEA boundaries are parallel arcs. For OEA construction, the RF start point is the radius extended to early ATT and the RF end point is the radius extended to late ATT.

(a) *Step 1*. Determine the leg R that is required to fit the geometry of the terrain/airspace. Enter this R value into formula 1-2-11 to verify the calculated bank angle (rounded to the nearest whole degree) is ≤ 25 degrees.

(b) *Step 2*. Locate the turn center at a perpendicular distance R from the inbound and outbound legs.

(c) *Step 3*. Construct an arc of radius R from the point on centerline tangent to the inbound leg at the turn fix to point on centerline tangent to the outbound leg at the turn fix. [Refer to paragraph 1-2-5.a(3) for alignment tolerance.]

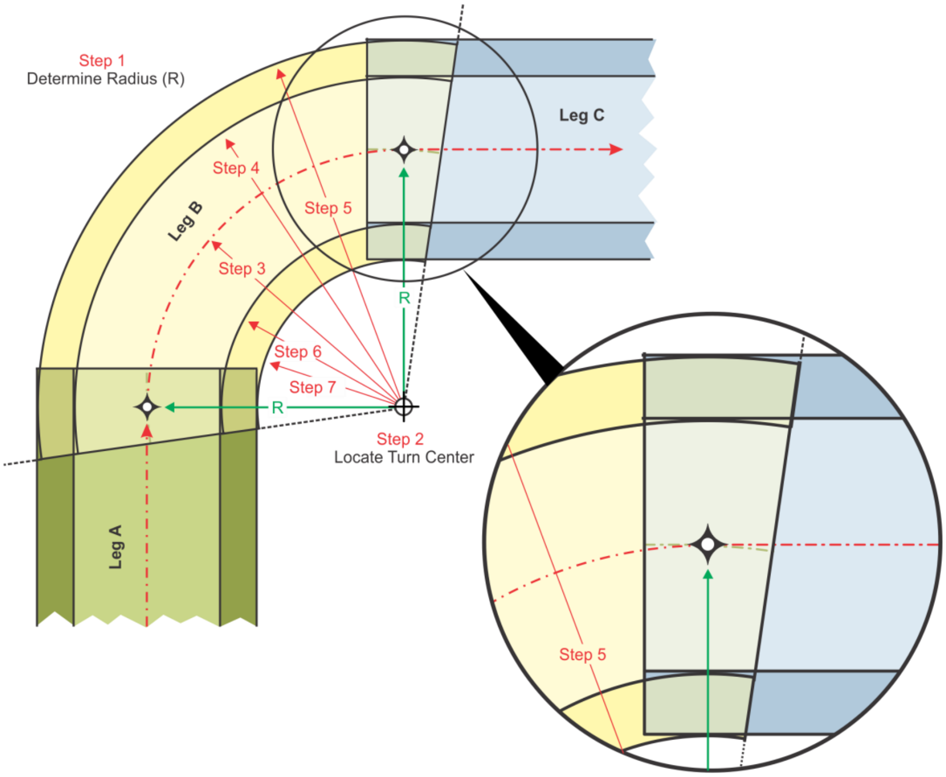
(d) *Step 4*. Primary area outer boundary. Construct an arc of radius R + primary area half width from a line originating at the turn center extending radially through the RF start point to a line originating at the turn center extending radially through the RF end point.

(e) *Step 5*. Secondary area outer boundary (if applicable). Construct an arc of radius R + primary area half width + secondary area width from a line originating at the turn center extending radially through the RF start point to a line originating at the turn center extending radially through the RF end point.

(f) *Step 6*. Primary area inner boundary. Construct an arc of radius R – primary area half width from a line originating at the turn center extending radially through the RF start point to a line originating at the turn center extending radially through RF end point.

(g) *Step 7*. Secondary area inner boundary (if applicable). Construct an arc of radius R – (primary area half width + secondary area width) from a line originating at the turn center extending radially through the RF start point to a line originating at the turn center extending radially through the RF end point.

Figure 1-2-10 RF Turn Construction



Section 1-3. Common Criteria

1-3-1. Common criteria. The following common requirements are applicable to PBN IFPs.

a. Descent gradient (DG) calculation. The applicable section of Order 8260.3, chapter 2 applies, except use formula 1-3-1 to calculate descent gradient.

Formula 1-3-1. Descent Gradient

$$\mathrm{DG} = \ln\left(\frac{\mathrm{r} + \mathrm{alt_b}}{\mathrm{r} + \mathrm{alt_e}}\right) \times \frac{\mathrm{r}}{\mathrm{D}}$$

Where:
$\mathrm{alt_b}$ = beginning altitude (feet)
$\mathrm{alt_e}$ = ending altitude (feet)
D = distance (NM) between the points of interest

Example:

$$\mathrm{DG} = \ln\left(\frac{\mathrm{r} + 6500}{\mathrm{r} + 3200}\right) \times \frac{\mathrm{r}}{10.36}$$

$$\mathrm{DG} \approx 318.46\ \mathrm{ft/NM}$$

b. Secondary area (when applicable). Obstacles located in secondary areas are subject to less obstacle clearance than those contained in the primary area. Unless otherwise specified, calculate the secondary ROC using formula 1-3-2.

Formula 1-3-2. Secondary ROC

$$\mathrm{ROC_{secondary}} = (\mathrm{ROC_{min}} + \mathrm{adj}) \times \left(1 - \frac{\mathrm{d_{primary}}}{\mathrm{w_s}}\right) + \mathrm{RASS}$$

Where:
$\mathrm{ROC_{min}}$ = min. secondary ROC (feet) at primary boundary [see Order 8260.3 chapter 2]
adj = ROC adjustments (feet) such as excessive length of final and precipitous terrain
RASS = remote altimeter setting source adjustment (feet) if applicable
$\mathrm{d_{primary}}$ = dist. (feet) from primary area boundary to obstacle. Zero (0) if not in secondary
$\mathrm{w_s}$ = secondary area width (feet)

Example:

$$\mathrm{ROC_{secondary}} = (250 + 74.32) \times \left(1 - \frac{405.86}{1822.83}\right) + 91.69$$

$$\mathrm{ROC_{secondary}} \approx 343.80\ \text{feet}$$

c. Turns to join initial and intermediate segments. The first leg of an initial or intermediate segment must be a TF that accommodates a 90-degree intercept angle. Use standard turn parameters at the start fix, except a 25-degree bank angle applies. Where a shorter leg is needed, reduce airspeed in increments of not less than five KIAS until the desired length is achieved [see table 1-2-2].

d. Initial approach segment. Initial approach segments consist of one or more TF or RF legs or a hold-in-lieu-of-procedure turn (HILPT). Standard length and alignment applies [see paragraph 1-2-5]. In some cases, initial segments are optional [see Order 8260.3, paragraph 2-4-1.a].

(1) Configuration. The default design incorporates a basic T configuration. The optional Basic Y configuration or RF leg initials may be used where beneficial.

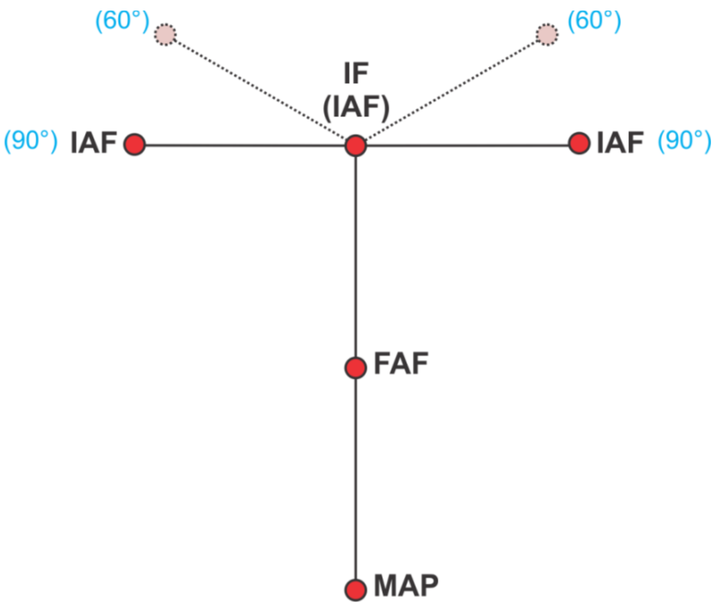
(a) Basic T. Two TF leg initial segments intercept the intermediate segment at 90-degree angles.

(b) Basic Y. One or both of the initial segments intercept at an angle less than 90 degrees, but normally not less than 60 degrees.

(c) Course reversal. Basic T/Y configurations normally include a HILPT course reversal initial established at a fix designated as an intermediate fix (IF)/Initial Approach Fix (IAF).

(d) Modified T/Y. Where not practical (e.g., maximum DG exceeded) one or both of the T/Y segments and/or the course reversal may be eliminated. Intercept angles less than 60 degrees and/or more than two segments may also be established where beneficial.

Figure 1-3-1. Basic T/Y Configuration



(2) Holding pattern. A holding pattern may be incorporated into the initial segment procedure design where an operational benefit can be derived; e.g., arrival holding at an IAF, course reversal pattern at the IF, etc. See Order 8260.3, chapter 17 for RNAV holding pattern construction guidance.

(a) Arrival holding. Ideally, the holding pattern inbound course should be aligned with the subsequent TF leg (tangent to course at the initial fix of the subsequent RF leg). If the pattern is offset from the subsequent TF leg course, the subsequent leg length must accommodate the resulting DTA requirement. The published holding altitude must be at or above the IAF altitude.

(b) Hold-in-lieu-of-procedure turn (HILPT). Ideally, establish the minimum holding altitude at the minimum IF fix altitude. In any case, the published holding altitude must result in a suitable descent gradient in the intermediate segment. If the pattern is offset from the subsequent TF leg course, the subsequent leg length must accommodate the resulting DTA requirement.

e. Intermediate approach segment. Intermediate approach segments consist of one or more TF or RF legs. Standard length and alignment rules apply [see paragraph 1-2-5] except as follows for the leg ending at the PFAF (the last leg):

(1) Alignment. Use the smallest amount of course change necessary. No course change is default.

(2) Length. For PBN SIAPs, intermediate segment length is independent of the amount of turn at the IF. The default length for CAT A/B is 3 NM. The default length for CAT C/D is 5 NM. The minimum length is as specified in paragraph 1-2-5.b(1).

f. Final approach segment. SIAPS designed to support Simultaneous Offset Instrument Approaches (SOIA) operations are exempted from standard final segment alignment when compliant with Order 8260.3, paragraph 16-3-5.

(1) Fixes.

(a) Final segment fixes. Establish stepdown fixes and VDPs as ATD fixes relative to the MAP.

(b) Obstacles close to a final approach or stepdown fix. Where turns are designed at the PFAF, Order 8260.3, chapter 2 “General Criteria” applies except for FB turns at the PFAF the 7:1 OIS starts ATT prior to the angle bisector, and extends 1 NM parallel to the final approach centerline.

(2) Final segment vertical path. By default, PBN IFPs should be designed so that all vertical guidance to the runway is based on common TCH and angles. All vertically-guided PBN approaches charted on the same IFP plate must share common TCH and GPA.

(a) GPA/VDA [see Order 8260.3, section 2-6].

(b) Threshold crossing height (TCH). The published TCH (nearest whole foot) should accommodate the largest aircraft height group normally expected to use the runway and must not be less than the minimum or exceed the maximum TCH. The TCH is based on achieving an acceptable wheel crossing height (WCH). The WCH is the difference between the published TCH and the approximate glidepath-to-wheel height [see table 1-3-1].

1. The default TCH provides a 30-foot WCH. The minimum WCH is 20 feet and the maximum WCH is 50 feet.

2. Displaced threshold. The TCH over a displaced threshold may result in a WCH of not less than 10 feet provided:

a Pavement equivalent to the strength of the landing runway is present prior to the displaced threshold.

b The calculated height of the glidepath over the beginning of the pavement prior to the displaced threshold is within the minimum/maximum TCH values.

(c) Vertical path calculations.

1. DA. Calculate the DA distance from the LTP/FTP using formula 1-3-3. For LPV/GLS, see paragraph 3-4-5.

2. PFAF location. To determine the PFAF location, calculate the distance from the LTP/FTP (straight-in aligned procedures) or MAP (circling aligned procedures) to the point the GPA/VDA or design angle intercepts the intermediate segment altitude [see formula 1-3-3]. For straight-in procedures the beginning altitude is LTP/FTP + TCH. For circling aligned procedures the beginning altitude is the airport elevation + 50.

Table 1-3-1. TCH Requirements

Representative Aircraft Type	Glidepath-to- Wheel Height (approximate)	Recommended TCH	Remarks
HEIGHT GROUP 1 General Aviation, Small Commuters, Corporate Turbojets, T-38, C-12, C-20, C-21, T-1, Fighter Jets, UC-35, T-3, T-6	10 feet or less	40 feet	Normally runways <6,000 long with reduced widths and/ or limited weight bearing, limiting larger aircraft use.
HEIGHT GROUP 2 F-28, B-737, C-9, DC-9, C-130, T-43, B-2	15 feet	45 feet	Regional airport with limited air carrier service.
HEIGHT GROUP 3 B-727/707/720/757, B-52, C-135, C-141, C-17, E-3, P-3, E-8, C-32	20 feet	50 feet	Runways not normally used by aircraft with glidepath-to-wheel heights > 20 feet.
HEIGHT GROUP 4 B-747/767/777, DC-10, A-300, B-1, KC-10, E-4, C-5, VC-25	25 feet	55 feet	Most primary runways at major airports.

Note: To determine the minimum allowable TCH, add 20 feet to the glidepath-to-wheel height and to determine the maximum allowable TCH, add 50 feet to the glidepath-to-wheel height.

Formula 1-3-3. Distance on Baro Glidepath

$$d_{Baro} = \ln\left(\frac{r + alt_e}{r + alt_b}\right) \times \frac{r}{\tan(\theta)}$$

Where:
alt_e = ending altitude (feet)
alt_b = beginning altitude (feet)

Example:

$$d_{Baro} = \ln\left(\frac{r + 4500}{r + 1441.6 + 47}\right) \times \frac{r}{\tan(3)}$$
$$d_{Baro} \approx 57452.70 \text{ feet}$$

3. Glidepath altitude. Calculate the altitude of the glidepath at any distance from the LTP/FTP using formula 1-3-4.

Formula 1-3-4. Altitude on Baro Glidepath

$$Z_{Baro} = e^{\left(\frac{d \times \tan(\theta)}{r}\right)} \times (r + alt_b) - r$$

Where:
alt_b = beginning altitude (feet)
d = distance (feet) between the points of interest

Example:

$$Z_{Baro} = e^{\left(\frac{4708.89 \times \tan(3)}{r}\right)} \times (r + 404.6 + 45) - r$$
$$Z_{Baro} \approx 696.39 \text{ feet}$$

g. Missed approach and departure.

(1) Sloping OCS. The OCS slope is dependent on the starting OCS elevation and a specific obstacle of interest. When calculating the height of the sloping surface, a default slope of 40:1 may be used to determine potential controlling obstructions. The actual OCS slope associated with any given obstacle is calculated using formula 1-3-5 and the OCS elevation at a point of interest can be found using formula 1-3-6.

Formula 1-3-5. OCS Slope

$$\mathrm{OCS_{slope}} = \frac{\mathrm{d_O}}{\ln\left(\frac{\mathrm{r + O_{elev}}}{\mathrm{r + OCS_{start}}}\right) \times \mathrm{r}}$$

Where:
$\mathrm{O_{elev}}$ = actual or $\mathrm{O_{adjusted}}$ obstacle elevation (feet)
$\mathrm{OCS_{start}}$ = starting altitude (feet)of the OCS
$\mathrm{d_O}$ = dist. (feet) from $\mathrm{AC_{start}}$ to obstacle (primary) or point abeam obstacle (secondary)

Example:

$$\mathrm{OCS_{slope}} = \frac{6076.12}{\ln\left(\frac{\mathrm{r} + 1152.6}{\mathrm{r} + 1000.6}\right) \times \mathrm{r}}$$

$$\mathrm{OCS_{slope}} \approx 39.98$$

Formula 1-3-6. OCS Elevation

$$\mathrm{OCS_{elev}} = \mathrm{e}^{\left(\frac{\mathrm{d_{OCS}}}{\mathrm{r \times OCS_{slope}}}\right)} \times (\mathrm{r + OCS_{start}}) - \mathrm{r}$$

Where:
$\mathrm{d_{OCS}}$ = distance (feet)from $\mathrm{OCS_{start}}$ point to the point of interest
$\mathrm{OCS_{start}}$ = starting altitude (feet) of the OCS

Example:

$$\mathrm{OCS_{elev}} = \mathrm{e}^{\left(\frac{2591.8}{\mathrm{r} \times 28.5}\right)} \times (\mathrm{r} + 1191.75) - \mathrm{r}$$

$$\mathrm{OCS_{elev}} \approx 1282.70 \text{ feet}$$

(2) Climb gradient (CG). Obstacle clearance in a climbing segment is dependent on the aircraft maintaining a minimum CG to a specified altitude. The minimum CG (also referred to as standard CG) is 200 ft/NM. When one or more obstacle penetrates the default OCS, a single increased CG that provides clearance over all penetrations may be established. A CG greater than 425 ft/NM is not authorized. Calculate the minimum climb gradient required to achieve the termination altitude at a given distance using formula 1-3-7.The aircraft altitude at a point of interest can be found using formula 1-3-8 and the distance required to achieve an altitude at any CG can be found using formula 1-3-9.

Formula 1-3-7. Required Climb Gradient

$$\mathrm{CG} = \ln\left(\frac{\mathrm{r} + \mathrm{CG_{term}}}{\mathrm{r} + \mathrm{AC_{start}}}\right) \times \frac{\mathrm{r}}{\mathrm{D_O}}$$

Where:
$\mathrm{CG_{term}}$ = climb gradient termination altitude (feet)
$\mathrm{AC_{start}}$ = starting altitude (feet) of the aircraft (e.g., DER elevation or SOC)
$\mathrm{D_O}$ = dist. (NM) from $\mathrm{AC_{start}}$ to obstacle (primary) or point abeam obstacle (secondary)

Example:

$$\mathrm{CG} = \ln\left(\frac{\mathrm{r} + 2626.52}{\mathrm{r} + 1402}\right) \times \frac{\mathrm{r}}{3.95}$$

$$\mathrm{CG} \approx 309.98\ \mathrm{ft/NM}$$

Formula 1-3-8. Achieved Aircraft Altitude

$$\mathrm{alt_e} = \mathrm{e}^{\left(\frac{\mathrm{D} \times \mathrm{Vert_{path}}}{\mathrm{r}}\right)} \times (\mathrm{r} + \mathrm{alt_b}) - \mathrm{r}$$

Where:
D = distance (NM) between the points of interest
$\mathrm{Vert_{path}}$ = aircraft vertical path (ft/NM)
$\mathrm{alt_b}$ = beginning altitude (feet)

Example:

$$\mathrm{alt_e} = \mathrm{e}^{\left(\frac{0.76 \times 309.98}{\mathrm{r}}\right)} \times (\mathrm{r} + 320) - \mathrm{r}$$

$$\mathrm{alt_e} = 555.59\ \text{feet}$$

Formula 1-3-9. Climb Distance Between Altitudes

$$\mathrm{D_{climb}} = \ln\left(\frac{\mathrm{r} + \mathrm{alt_e}}{\mathrm{r} + \mathrm{alt_b}}\right) \times \frac{\mathrm{r}}{\mathrm{CG}}$$

Where:
$\mathrm{alt_e}$ = ending altitude (feet)
$\mathrm{alt_b}$ = beginning altitude (feet)

Example:

$$\mathrm{D_{climb}} = \ln\left(\frac{\mathrm{r} + 3300}{\mathrm{r} + 2105}\right) \times \frac{\mathrm{r}}{315}$$

$$\mathrm{D_{climb}} \approx 3.79\ \mathrm{NM}$$

(3) Wind spiral (WS) parameters. Use the following steps to determine the parameters used in WS construction.

(a) *Step 1*. Calculate KTAS with using the specified turn altitude/assumed fix altitude and applicable KIAS. Input this value as the ground speed in formula 1-2-10 to find the no-wind turn radius.

(b) *Step 2*. Calculate the Turn Rate (TR) using formula 1-3-10. Maximum TR is three degrees per second.

Formula 1-3-10. Turn Rate (TR)

$$\mathrm{TR} = \text{lesser of 3 or } \frac{180 \times \mathrm{V_{KTAS}}}{\pi \times \mathrm{R} \times 3600}$$

Where:
$\mathrm{V_{KTAS}}$ = KTAS using specified turn alt./assumed fix alt. and applicable KIAS
R = calculated no-wind turn radius (NM)

Example:

$$\mathrm{TR} = \frac{180 \times 281.30}{\pi \times 4.30 \times 3600}$$
$$\mathrm{TR} \approx 1.04\ \mathrm{degrees/second}$$

(c) *Step 3*. Determine total magnitude of turn.

1. Turn-at-altitude. Calculate the total turn magnitude by measuring the arc distance (in degrees) along the no-wind turn radius from the course centerline/late-turn baseline intersection to the direct-to-fix line tangent point.

2. Turn at FO fix. Where the next leg is a DF, use turn-at-altitude methodology. For TF legs, calculate the total turn magnitude by comparing the inbound and outbound tracks.

(d) *Step 4*. Determine the highest altitude in the turn using the applicable vertical path [see paragraph 1-2-5.c(1)(a)4] starting at the specified turn altitude/assumed fix altitude plus 1 NM of flight track distance per 12 degrees total turn magnitude.

(e) *Step 5*. Calculate the wind spiral increase (ΔR) relative to the no-wind turn radius for a given incremental turn magnitude using formula 1-3-11. The incremental turn magnitude is an angular measurement from the wind spiral baseline about the no-wind turn radius center point which ranges from zero to the total turn magnitude.

Formula 1-3-11. Wind Spiral Δ*R*

$$\Delta\mathrm{R} = \frac{\mathrm{V}_{\mathrm{KTW}} \times \beta_{\mathrm{increment}}}{3600 \times \mathrm{TR}}$$

Where:
$\beta_{\mathrm{increment}}$ = incremental turn magnitude at point of interest
$\mathrm{V}_{\mathrm{KTW}}$ = calculated or historical tailwind (knots) using the highest altitude in the turn
$\mathrm{TR} =$ calculated turn rate (degrees/second)

Example:

$$\Delta\mathrm{R} = \frac{51.55 \times 35}{3600 \times 1.04}$$
$$\Delta\mathrm{R} \approx 0.48\ \mathrm{NM}$$

(4) Construction baselines. Missed approach/departure construction baselines are commonly identified by the points that they are comprised of, which can be used to mark the beginning, ending, and/or tie-back point for the OEA. When these lines are used to identify tie-back points for turn construction, the placement of the point will be in alphabetical order from turn-side to non-turn side. For example, a right turning missed approach with lines C-D and L-L' will have points C and L on the right, while a left turning missed approach will have them on the left.

THIS PAGE IS INTENTIONALLY LEFT BLANK

Chapter 2. En Route, Standard Terminal Arrivals, Feeder Routes, and Terminal Arrival Areas (TAA).

Section 2-1. En Route

2-1-1. Air Traffic Service (ATS) Q/T routes. Q and T routes may be developed to support en route PBN operations. XTT is RNAV 2 en route flight phase. Secondary areas apply.

a. General.

(1) Q routes are for use from 18000 feet MSL through FL450 inclusive.

(2) T routes are for use from as low as 1200 feet AGL up to but not including 18000 feet MSL.

(3) Leg/Fix type. Use TF legs only. Fixes may only be FB waypoints or NAVAIDs.

b. OEA construction.

(1) Area. Except for route termination fixes, apply paragraph 1-2-5. Where applicable, construction is bi-directional.

(2) Construction at route termination fixes. Order 8260.3 paragraph 15-1-2 applies except primary and secondary boundary arcs are centered ATT past the termination fix.

c. Obstacle clearance. Order 8260.3, section 15-2 applies. Secondary ROC may be calculated as specified in Order 8260.3, section 15-2 or formula 1-3-2 (of this order).

d. Altitudes. Order 8260.3, section 15-3 applies except MCA distances are measured from ATT past the fix.

Section 2-2. Standard Terminal Arrival (STAR)

2-2-1. PBN STAR. A PBN STAR may be established to transition from the en route environment to the terminal environment. For the purposes of OEA construction, the default XTT is RNAV 1, STAR/Feeder/TAA flight phase. Optional XTT is RNP 1 which applies when RF legs are used. For STARs that serve multiple airports, use the most conservative distance to determine where the effective XTT is RNAV/RNP 2 [see table 1-2-1 footnote 2]. Secondary areas apply. Order 8260.3, section 2-2 applies in conjunction with the following PBN specific criteria.

a. General.

(1) Leg type. Use TF or RF legs, except a heading to a manual termination (VM) or from a fix to a manual termination (FM) leg may be established at the STAR termination.

(2) Fixes.

(a) All fixes prior to the termination fix must be FB.

(b) The termination fix may be either FB or FO depending on whether the STAR connects to a SIAP, the attributes of the common fix, and leg type. For this purpose, the first coded fix of the SIAP (i.e., the start of the SIAP "IF" leg) is considered a FB fix.

1. When the termination fix is a common fix followed by an FM or VM leg, the termination fix attribute must be FO and the SIAP fix should be FB.

2. For other cases, common fix attributes must be identical.

3. When the termination is not at a common fix, the STAR must terminate with a VM or FM leg at a FO fix.

Note: When more than one transition/common route terminates at the same fix, more than one of the above can apply simultaneously**.** In those cases, the waypoint attribute must be the same on each of the common routes/runway transitions.

b. Area. Apply paragraph 1-2-5 for area construction and alignment.

Note: When the STAR terminates at a fix on a SIAP, paragraph 1-2-5 applies between the last leg of a STAR and the first leg of SIAP. When a STAR terminates in a VM or FM leg ensure the fix altitude is at or above the MVA/MIA.

c. Obstacle clearance. Apply Order 8260.3, paragraph 2-2-1. Alternatively, formula 1-3-2 (of this order) may be used to calculate secondary ROC.

Section 2-3. Feeder Routes

2-3-1. Feeder route. A feeder route may be developed as an optional transition from the en route environment to the terminal structure [see Order 8260.19, paragraph 4-6-2]. Use TF and/or RF legs. Default XTT is RNAV 1, STAR/Feeder/TAA flight phase. Optional XTT is RNP 1 and applies to any feeder route incorporating an RF leg. Secondary areas apply.

a. General. Apply paragraph 1-2-5 for course alignment.

b. Area. Apply paragraph 1-2-5 for area construction.

c. Obstacle clearance. Apply Order 8260.3, paragraph 2-3-1. Alternatively, formula 1-3-2 (of this order) may be used to calculate secondary ROC.

Section 2-4. Terminal Arrival Area (TAA)

2-4-1. General. A TAA may be established as an optional feeder/arrival transition from the en route structure. It consists of TAA arrival areas developed in conjunction with an approach procedure utilizing RNAV initial segments. In the TAA arrival areas, the PBN is RNAV 1 STAR/Feeder/TAA flight phase. Secondary areas are not applied; however, buffers are defined from each area boundary.

2-4-2. Approach procedure. Design in accordance with the applicable criteria in conjunction with the following guidance.

a. Initial approach segments.

(1) Use TF legs (FB fixes only) in a basic or modified T configuration only [see paragraph 1-3-1.d].

(2) Maximum leg length is 15 NM.

(3) A hold-in-lieu-of-procedure turn (HILPT) initial segment must be established at the IF/IAF at an altitude compatible with the TAA arrival area altitudes [see paragraph 2-4-3.e]. The inbound holding course must be aligned with the intermediate segment.

b. Intermediate approach segment. Establish a single dual purpose IF/IAF using TF (FB fixes only) aligned with the FAS.

c. Missed approach segment. Where possible, design missed approach segments to allow a “direct entry” into a missed approach holding pattern [see figure 2-4-1]. If the missed approach routing terminates at a base area IAF, align the missed approach holding pattern with the initial inbound course [see figure 2-4-2].

Figure 2-4-1. Default Missed Approach Holding Alignment

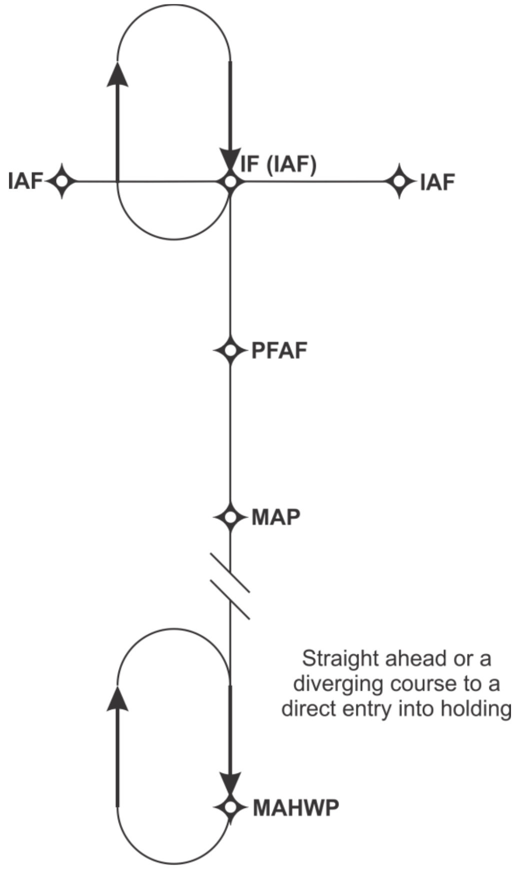
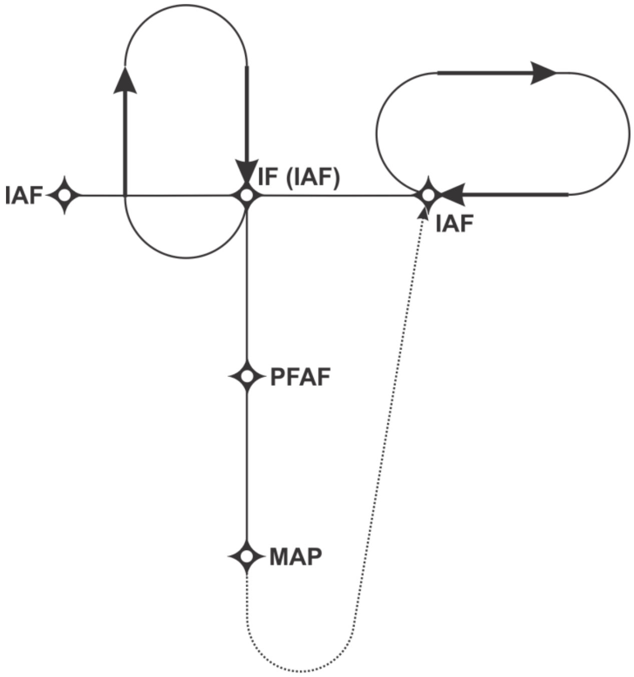
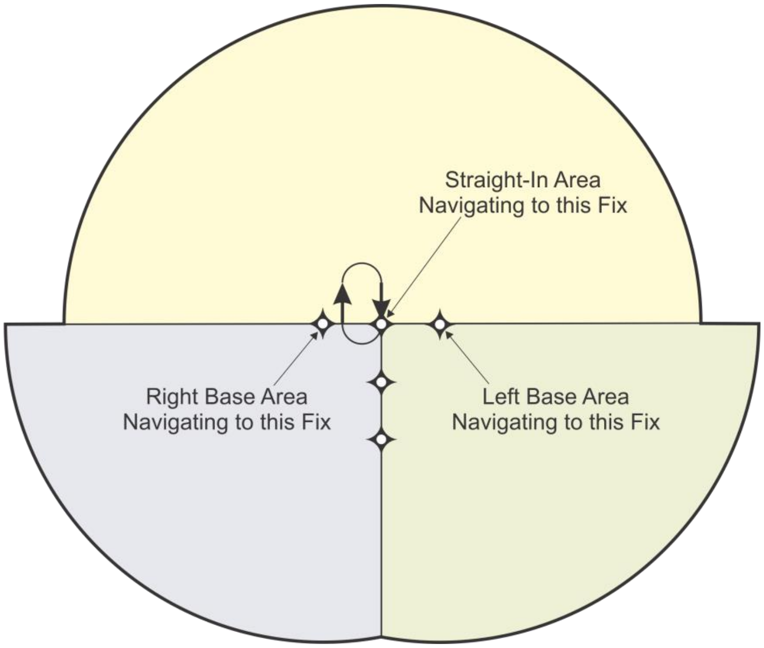

IF (IAF)
IAF
IAF
PFAF
MAP
Straight ahead or a diverging course to a direct entry into holding
MAHWP

Figure 2-4-2. Missed Approach Holding Alignment at T IAF



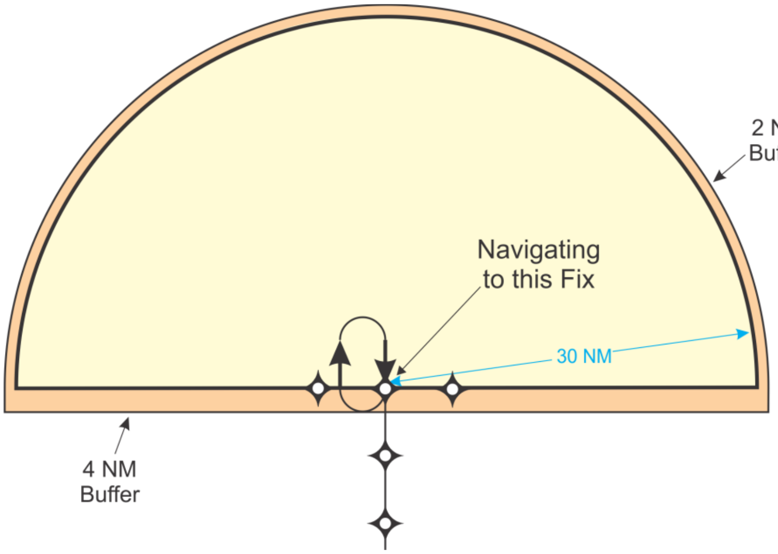
2-4-3. Arrival areas. A default TAA contains a straight-in area, a right base area, and a left base area. Each area is defined relative to a specific IAF [see figure 2-4-3]. The boundary between the straight-in area and the base areas is defined by extensions of the T initial segment track centerlines. The boundary between the right and left base areas is an extension of the intermediate segment track centerline.

Figure 2-4-3. Default TAA Arrival Areas



a. Straight-in area. Construct the base area side boundary by extending a 30-NM line from the IF/IAF through each T initial segment IAF. Construct the outside boundary by connecting the base area side boundary lines with a 30-NM arc centered on the IF/IAF [see figure 2-4-4]. Apply a 4-NM obstacle evaluation buffer on the base area side boundary and a 2-NM buffer on the outside boundary.

Figure 2-4-4. Straight-in Area



b. Right and left base area. The straight-in side boundaries are the initial segment track centerlines from IAF/IF to each IAF extended outward 30 NM. The boundary between the right and left base areas (inside boundary) is an extension of the intermediate segment track centerline. Construct the outside boundaries as 30-NM arcs centered on each IAF from the straight -in side boundary to in the appropriate direction (clockwise or counter-clockwise) to the intersection of the intermediate segment track centerline extended [see figure 2-4-5]. Apply a 4-NM obstacle evaluation buffer on the straight-in and inside boundaries and a 2-NM buffer on the outside boundary.

Figure 2-4-5. Right Base Area

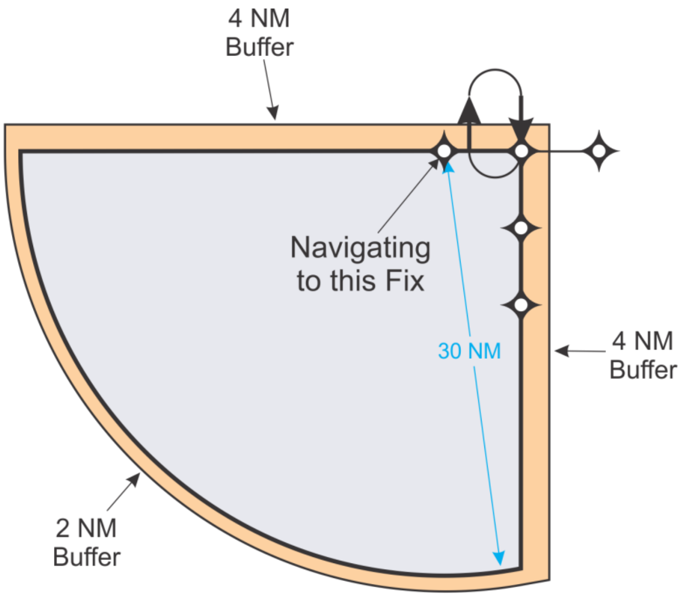
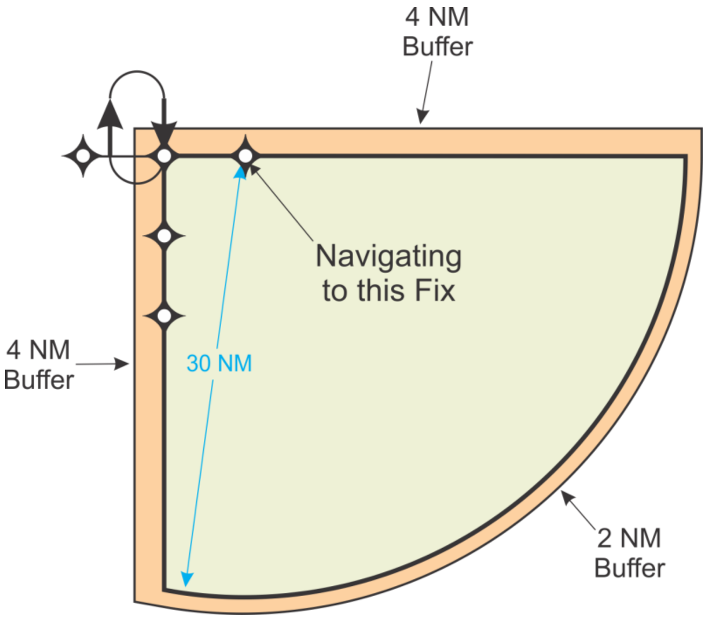


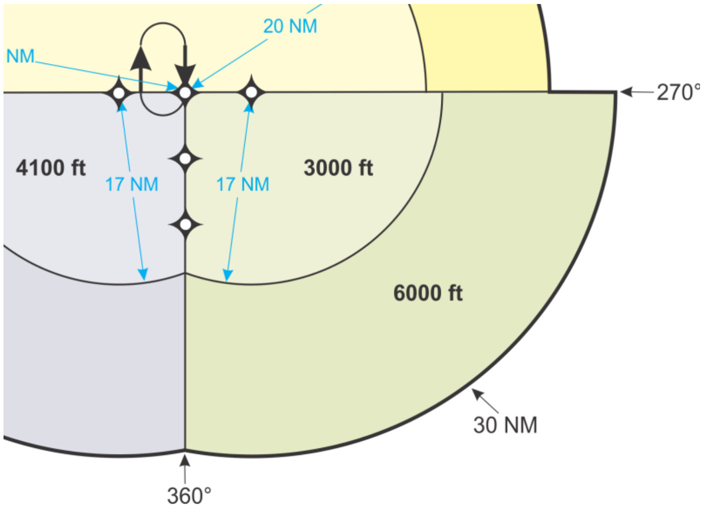
Figure 2-4-6. Left Base Area



c. Area modification. Modifications to the default TAA design may be necessary to achieve an acceptable descent gradient or to meet operational requirements.

(1) Stepdown arcs. Areas may be sub-divided using a stepdown arc [see figure 2-4-7]. Stepdown arcs are centered on the fix associated with the area and may not be less than 4 NM from the center fix or within 4 NM of the outer boundary. Only one stepdown arc may be established per area/sector. A 2-NM obstacle evaluation buffer applies from the stepdown arc. The altitude at the stepdown arc must not result in an excessive descent gradient in the associated NoPT initial calculated as length plus stepdown arc radius.

Figure 2-4-7. Area Modification; Stepdown Arcs



(2) Radial sectorization. The straight in or extended straight-in area [see paragraph 2-4-3.c(3)(a)] may be divided into up to three sectors defined radially by inbound courses to the IF/IAF [see figure 2-4-8]. Radial sectors can be used alone or in conjunction with stepdown arcs [see figure 2-4-9]. The angular size of a sector may not be less than 30 degrees, except a sector with a side defined by a base area boundary that also includes a stepdown arc may not be less than 45 degrees. A 4-NM obstacle evaluation buffer applies to radial sector boundaries.

Figure 2-4-8. Area Modification; Straight-in Area Radial Sectorization

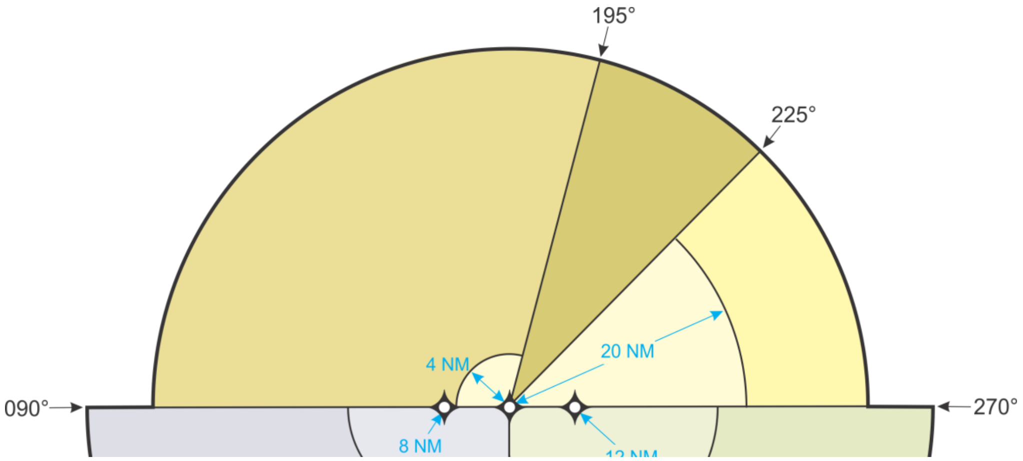
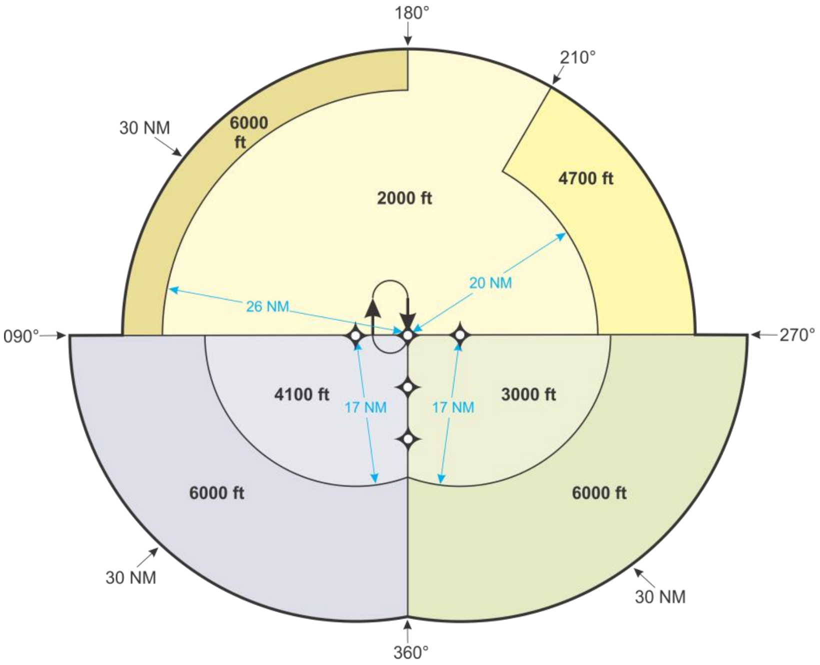


Figure 2-4-9. Area Modification; Radial Sectorization and Stepdown Arcs



(3) Base area elimination. One or both of the basic T initial approach segments and its associated base area may be eliminated [see figure 2-4-10 and figure 2-4-11].

(a) Construction. The straight-in area construction is extended into the area of the eliminated base area(s). Paragraph 2-4-3.b construction applies to a remaining base area. Paragraph 2-4-3.a applies except the outside boundary is a 30-NM arc connecting the base area side boundary line to the intermediate segment track centerline extended. When both base areas are eliminated, the straight-in area is a 30-NM circle centered on the IF/IAF.

Figure 2-4-10. Area Modification; Right Base Area Elimination

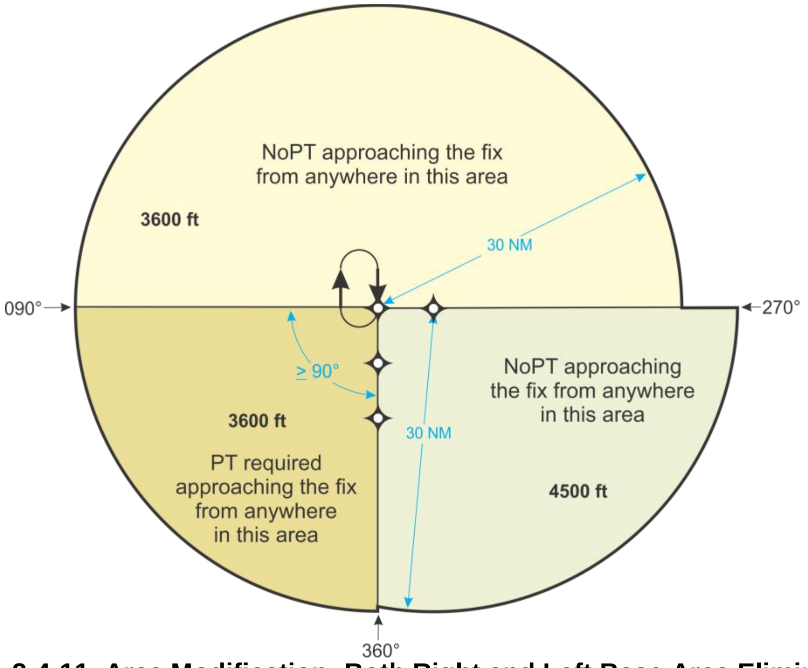
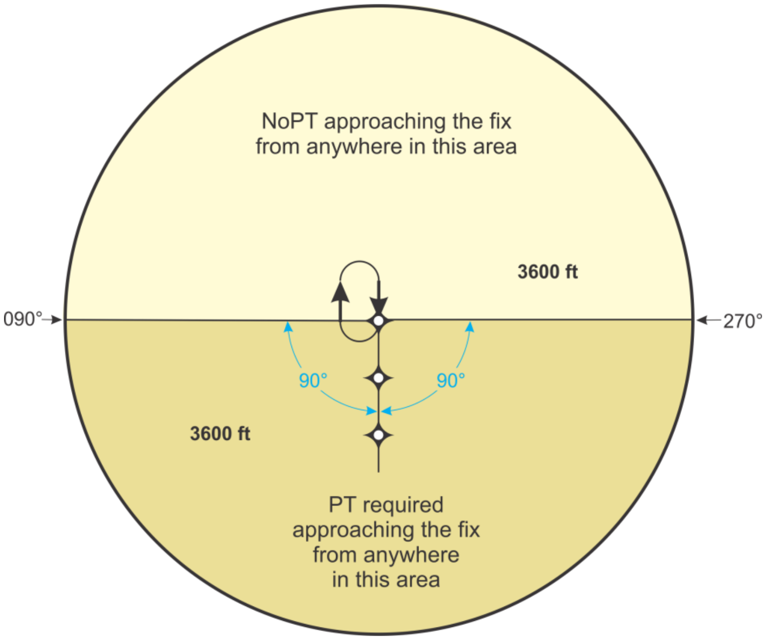
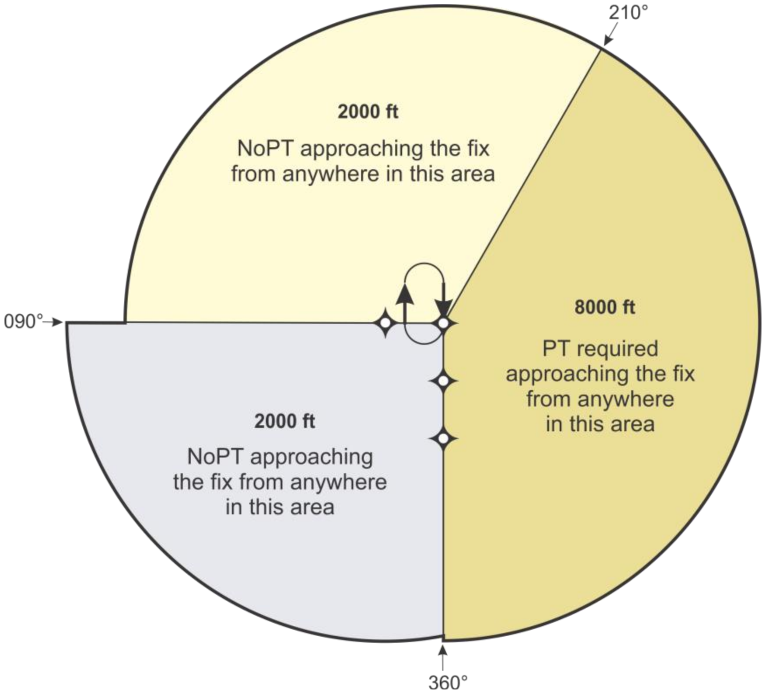


Figure 2-4-11. Area Modification; Both Right and Left Base Area Elimination



(b) No procedure (NoPT) exclusion. The sector associated with an eliminated base area must be excluded from a NoPT designation. A NoPT exclusion area may also extended from an eliminated base area into the straight-in area when required to meet descent gradient standards.

Figure 2-4-12. Area Modification; NoPT Exclusion

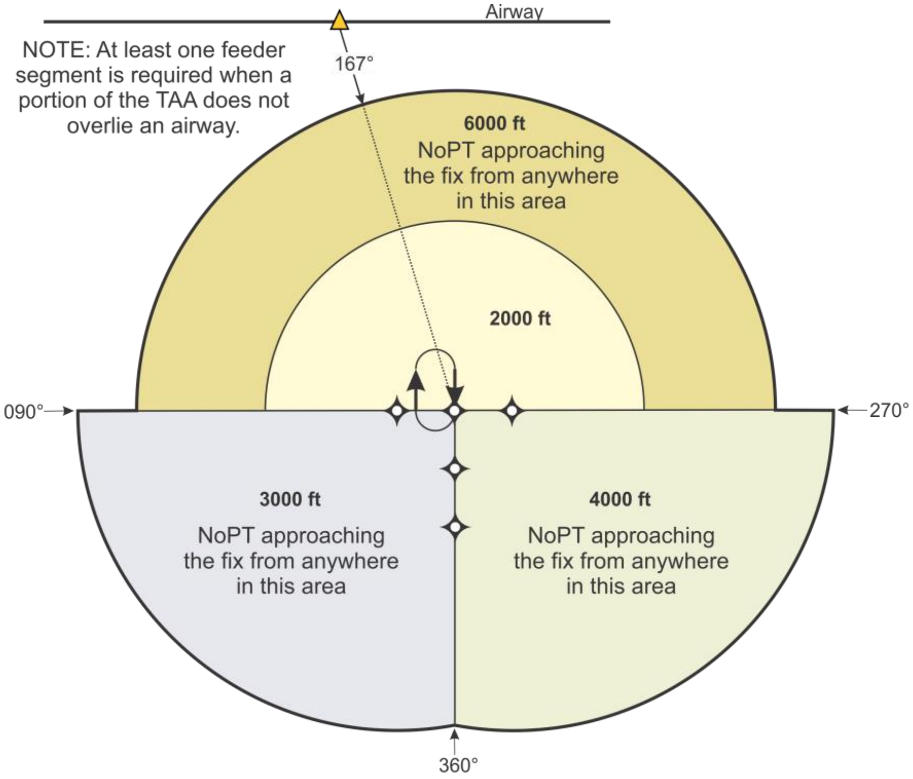

210°
2000 ft
NoPT approaching the fix from anywhere in this area
090°
8000 ft
PT required approaching the fix from anywhere in this area
2000 ft
NoPT approaching the fix from anywhere in this area
360°

d. Obstacle clearance [see Order 8260.3, paragraph 2-3-1.c].

e. Altitude selection. A minimum altitude is specified for each area and sub-sector. Where possible establish a common IAF altitude for the right/left base and straight-in areas. Where not possible, area altitudes must permit establishment of a common altitude at the IF/IAF. Except for areas/sub-sectors excluded from a NoPT designation, the associated IAF altitude may not be lower than the applicable area/sub-sector altitude. Order 8260.3, paragraph 2-4-5.e applies to area/sub-sectors excluded from a NoPT designation.

2-4-4. Feeder route. Normally one of the TAA areas (excluding obstacle evaluation buffers) will overlie an airway. Where this is not the case, at least one feeder route must be established from the en route structure to the TAA boundary. Order 8260.3, paragraph 2-3-1 applies except alignment must be a direct course to the IF/IAF or right/left base IAF depending on the location of the feeder fix [see figure 2-4-13].

Figure 2-4-13. TAA with Feeder route


Airway
NOTE: At least one feeder segment is required when a portion of the TAA does not overlie an airway.
167°
6000 ft
NoPT approaching the fix from anywhere in this area
2000 ft
090°
270°
3000 ft
NoPT approaching the fix from anywhere in this area
4000 ft
NoPT approaching the fix from anywhere in this area
360°

Chapter 3. Area Navigation (RNAV) Approach

Section 3-1. General Criteria

3-1-1. Applicability. This chapter applies to procedures covered by the RNP APCH NavSpec (LPV, LP, LNAV/VNAV, and LNAV). It also applies to PBN legs that transition to an ILS or GLS final. Design transitions to a GLS final with a final track intercept within 20 NM of the airport. See Order 8260.19, chapter 8 paragraph 'Minimums' for lines of RNAV minimums.

3-1-2. Feeder segment. Construct feeder segments as described in chapter 2 using one or more TF or RF legs. Segment XTT is RNAV 1 STAR/Feeder/TAA flight phase. Secondary areas apply.

3-1-3. Initial. Construct initial segments as described in chapter 1 using one or more TF or RF legs. Segment XTT is RNP APCH, Initial flight phase. Secondary areas apply.

3-1-4. Intermediate. Construct intermediate segments as described in chapter 1 using one or more TF or RF legs. Segment XTT is RNP APCH, Intermediate flight phase. Secondary areas apply. Paragraph 1-2-5.b(1)(d) applies except the ATT at the PFAF is based on the applicable final approach navigation accuracy from Table 1-2-1.

a. RF leg. Except for RNAV intermediate to an ILS final [see Appendix C], an RF leg must end at least 2 NM prior to the PFAF.

b. Alignment (maximum course change at the PFAF). Offset alignment is only authorized when the PFAF is a FB fix.

(1) LNAV and LP. Align the intermediate course within 30 degrees of the final approach course.

(2) LNAV/VNAV and LPV/GLS. Align the intermediate course within 15 degrees of the final approach course.

c. Area [see figure 3-1-1and figure 3-1-2].

(1) LNAV and LNAV/VNAV. The intermediate segment primary area tapers uniformly from ±2 NM at a point 2 NM prior to the PFAF to the primary boundary 1 NM past the PFAF. The secondary boundary tapers uniformly from 1 NM at a point 2 NM prior to the PFAF to the secondary boundary 1 NM past the PFAF.

(2) LP and LPV/GLS. The intermediate segment primary area tapers uniformly from ±2 NM at a point 2 NM prior to the PFAF to the primary boundary/X OCS outer boundary abeam the PFAF. The secondary boundary tapers uniformly from 1 NM at a point 2 NM prior to the PFAF to the secondary/Y OCS outer boundary abeam the PFAF.

Figure 3-1-1. LNAV, LNAV/VNAV Intermediate

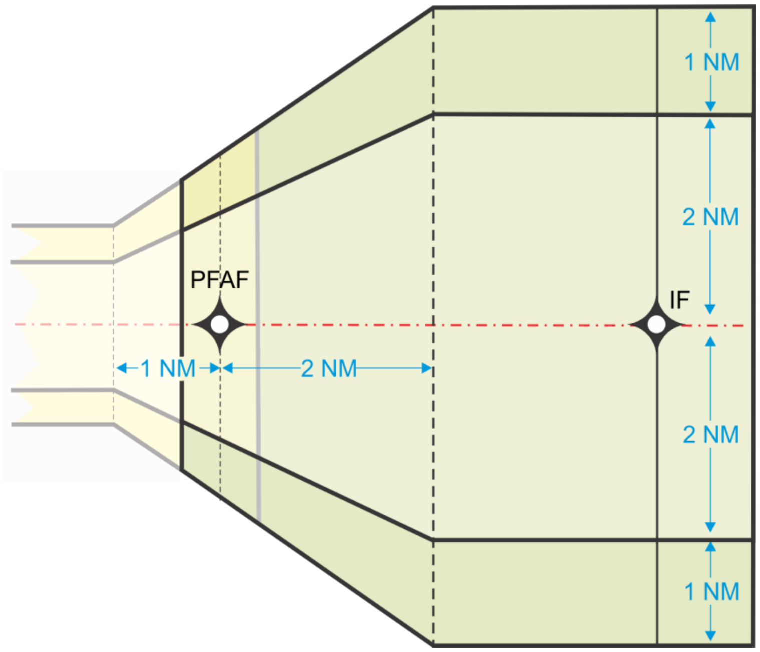
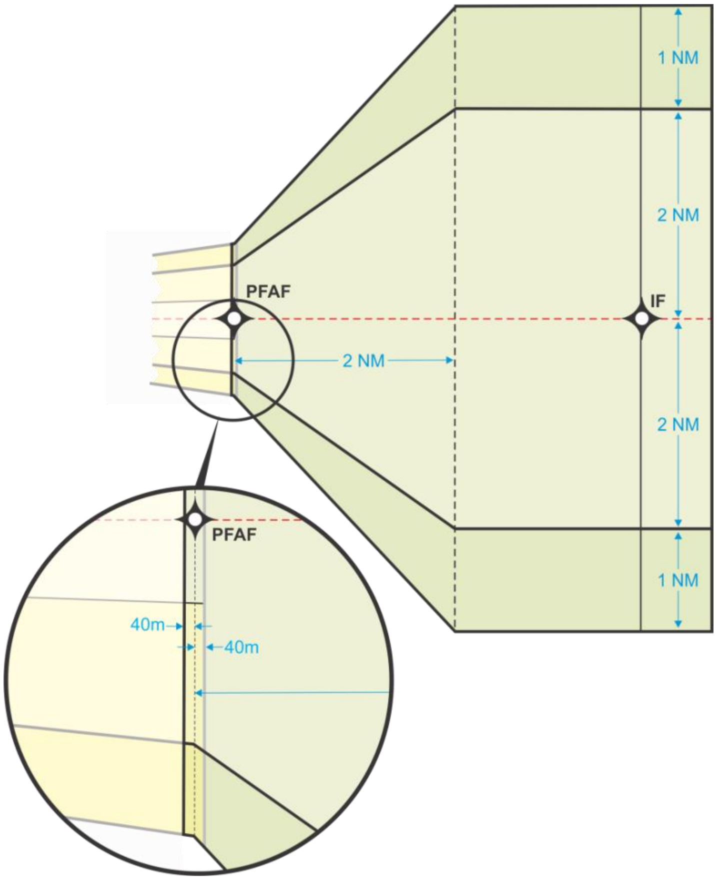


Figure 3-1-2. LP, LPV/GLS Intermediate



(3) Turn at IF or start of aligned leg prior to PFAF. When the standard FB turn construction results in boundaries outside the normal primary/secondary widths at the point 2 NM prior to the PFAF, the inside (turn side) boundaries may constructed as straight lines from the tangent points on the turning construction arcs to the applicable final segment boundaries [see figure 3-1-3 and figure 3-1-4]. Redesign if straight line connections result in an area less than normal primary/secondary area width in the leg prior to PFAF.

Figure 3-1-3. Turn at IF Example LNAV, LNAV/VNAV

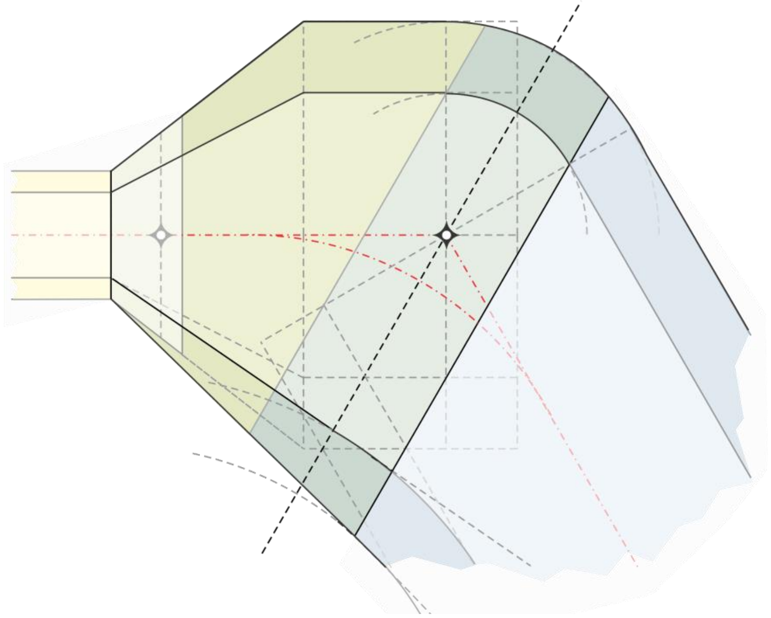
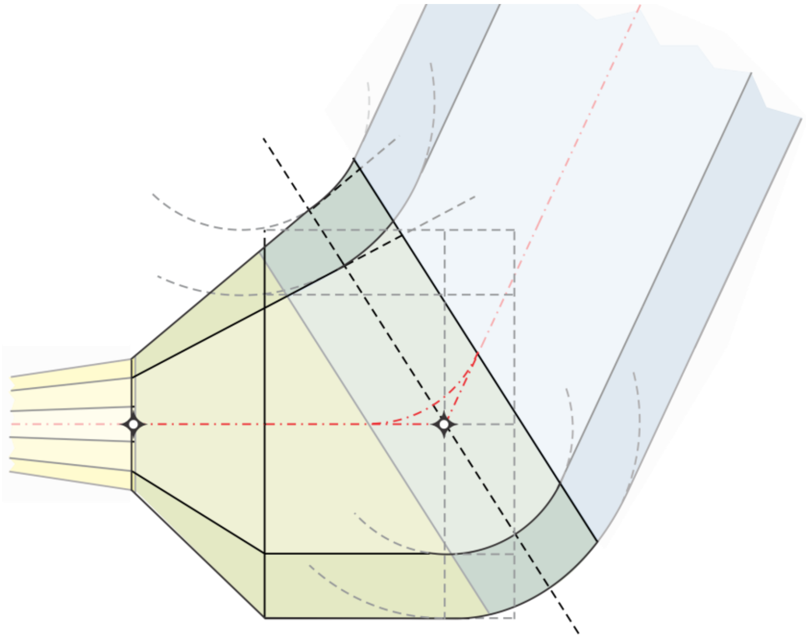


Figure 3-1-4. Turn at IF Example LP, LPV/GLS



(4) Offset construction. Where the intermediate course is not an extension of the final course, use the following construction techniques.

(a) *Step 1*. Construct line A perpendicular to the intermediate course 2 NM prior to the PFAF.

(b) *Step 2*. Construct line B perpendicular to the intermediate course extended 1 NM past the PFAF.

(c) *Step 3*. Construct the inside turn boundaries by connecting the points of intersection of line A with the turn side intermediate segment boundaries with the intersection of line B with the turn side final segment boundaries.

(d) *Step 4*. This step is dependent on procedure.

1. LNAV and LNAV/VNAV [see figure 3-1-5]. Construct arcs centered on the PFAF of 1 NM and 1.3-NM radius on the non-turn side of the fix.

2. LP and LPV/GLS [see figure 3-1-6]. Connect lines from the point of intersection of line A and the outside primary and secondary intermediate segment boundaries to the final segment primary and secondary final segment lines at a point perpendicular to the final course at the PFAF.

(e) *Step 5* (applicable to LNAV and LNAV/VNAV Only). Connect lines from the point of intersection of line A and the outside primary and secondary intermediate segment boundaries to tangent points on the arcs constructed by *Step 4*.

(f) *Step 6* (applicable to LNAV and LNAV/VNAV Only). Connect lines tangent to the arcs created in *Step 4* that taper inward at 30 degrees relative to the FAC to intersect the primary and secondary final segment boundaries as appropriate.

Figure 3-1-5. Offset Intermediate, Example LNAV

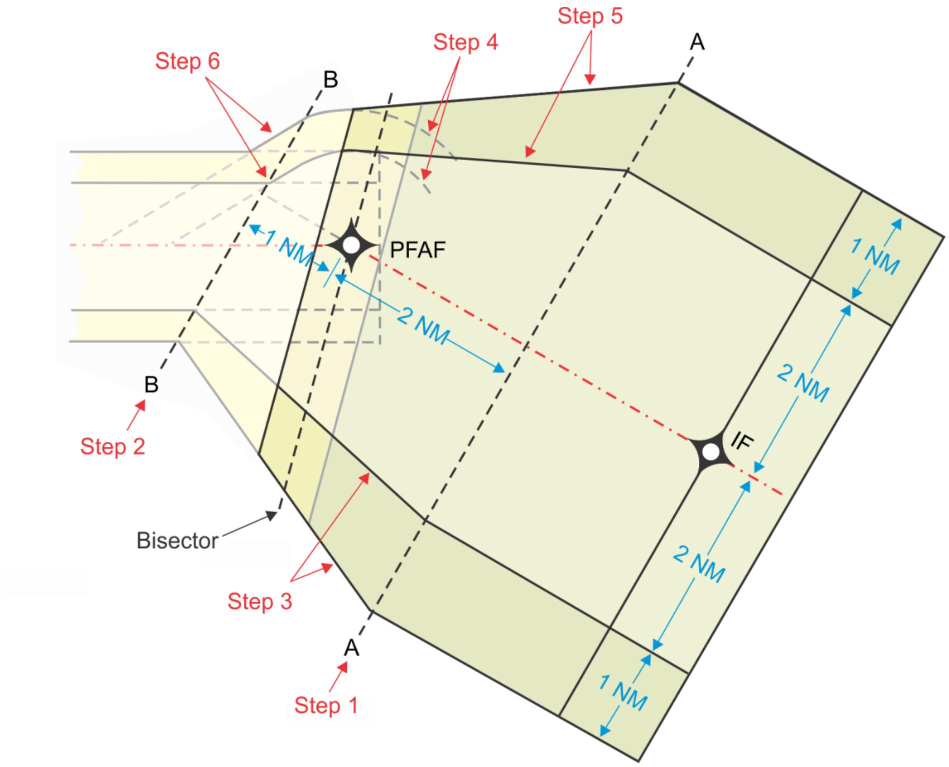
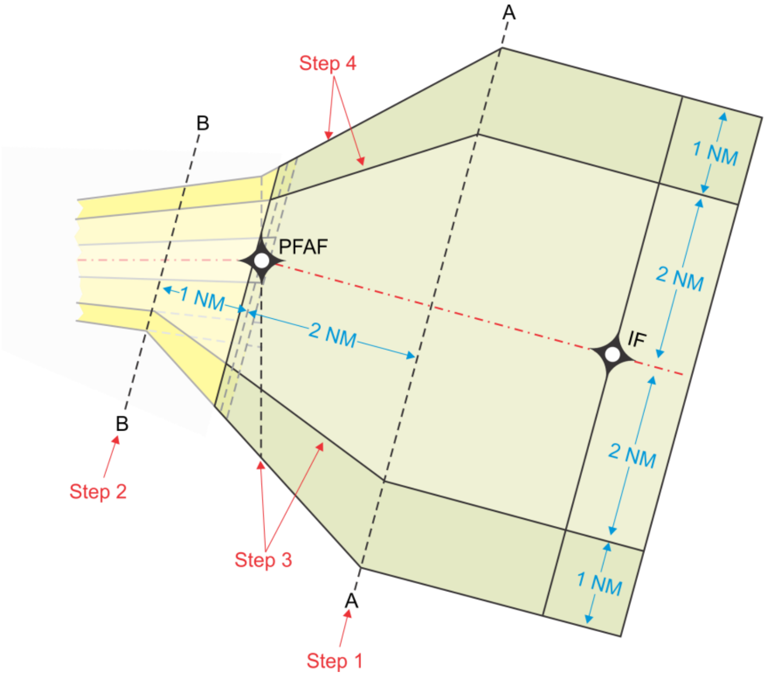


Figure 3-1-6. Offset Intermediate Example LPV/GLS



3-1-5. Final. Construct final segments as described in section 3-2 using one or more TF legs. RF legs and TF turns are not authorized. Segment XTT is RNP APCH, final flight phase (part A for LNAV and LNAV/VNAV, part B for LP and LPV/GLS). Secondary areas apply.

a. General. Paragraph 1-3-1.f applies except see paragraph 3-4-1 for LPV/GLS exceptions.

b. The final segment OEA primary and secondary boundaries follow the intermediate segment OEA primary and secondary boundaries from ATT prior to the PFAF to the applicable taper connection point and includes offset intermediate construction [see figure 3-1-1, figure 3-1-2, figure 3-1-3, figure 3-1-5, and figure 3-1-6].

c. FAS data block [see figure 3-1-7]. The FAS data block is documentation requirement used to define the lateral and vertical paths unique to SIAPs with LP, LPV, or GLS minimums. FAS data includes the following fields;

(1) Flight path alignment point (FPAP). The FPAP is defined by geodetic coordinates. When the FAC is aligned with the RCL, find the FPAP by extending a geodesic line from the LTP toward the DER a distance equal to the greater of the runway length or 9023 feet. For offset procedures, the FPAP is located on the extension of the desired final approach course at a distance from the FTP that provides the appropriate lateral course width.

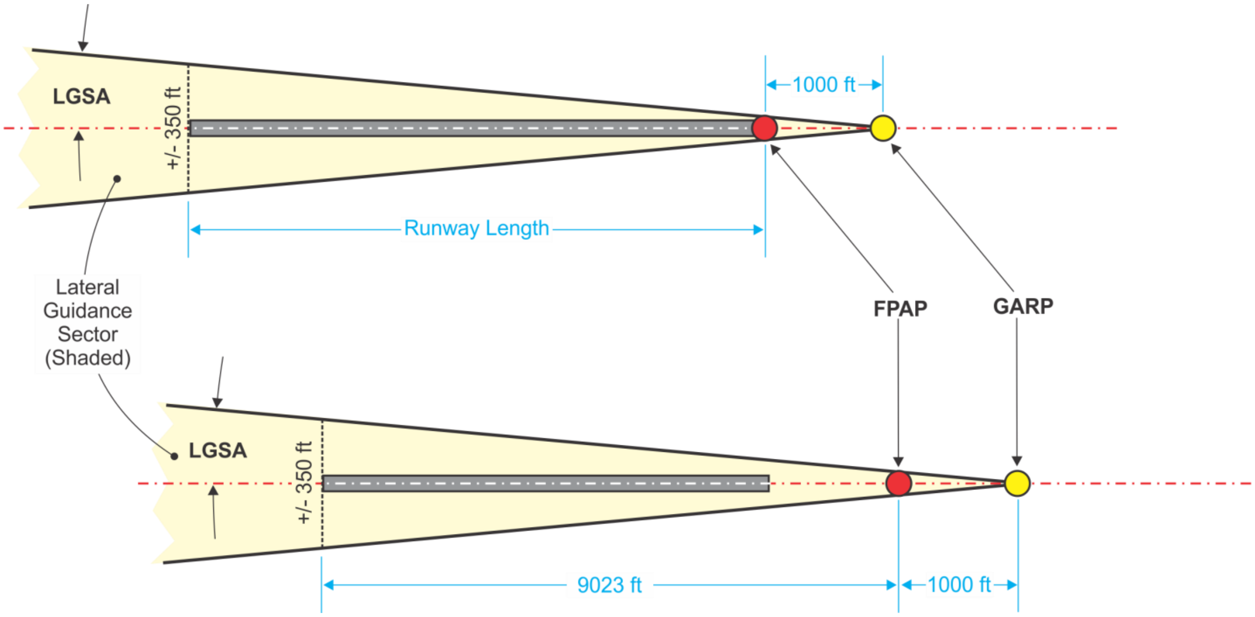
(2) Length offset. The length offset value is the distance between the departure end of runway and the FPAP, rounded to the nearest eight-meter value. It defines the location where lateral sensitivity changes to missed approach sensitivity. For offset procedures and where the FPAP is located at the departure end of the runway, length offset is zero (0).

(3) Course width at threshold. The course width at threshold is defined by the GNSS azimuth reference point (GARP) and the lateral guidance sector angle (LGSA) using formula 3-1-1. Convert the result to meters and round to the nearest 0.25-meter increment.

(a) GARP. The GARP is a point located 1000 feet beyond the FPAP on the extension of the FAC/RCL geodesic line. This point is used by the airborne system as the origin of the lateral guidance sector.

(b) LGSA. The LGSA originates at the GARP and is the angular dimension of lateral guidance relative to the FAC. The LGSA angle ranges from not more than ±2 degrees to not less than ±1.5 degrees.

Figure 3-1-7. FAS Data Block



Formula 3-1-1. Course Width At Threshold

$$\text{Course}_{\text{width}} = \text{greater of 350 feet or } \tan(1.5) \times d_{\text{GARP}}$$

Where:
d_{GARP} = distance (feet) from the LTP/FTP to the GARP

Example:

$$\text{Course}_{\text{width}} = \tan(1.5) \times 13560.66$$
$$\text{Course}_{\text{width}} \approx 355.10 \text{ feet}$$
$$\text{Course}_{\text{width}} = 108.25 \text{ meters}$$

Section 3-2. General Non-Vertically Guided Final Segment

3-2-1. General. This section contains obstacle evaluation criteria for LNAV and LP non-vertically guided approach procedures. LP is not applicable if straight-in minimums are not authorized [see paragraph 3-2-2.b and Order 8260.3 paragraph 2-6-2]. Do not develop a new LP to a runway where an LNAV/VNAV or LPV can be developed for at least one approach CAT, regardless of minima.

3-2-2. Alignment. Optimum non-vertically guided procedure final segment alignment is with the runway centerline extended through the LTP. When published in conjunction with a vertically guided procedure, alignment must be identical with the vertically guided final segment. TF-TF turns are not allowed in the FAS.

a. Offset. When the final course must be offset, it may be offset up to 30 degrees (published separately from vertically guided) when the following conditions are met:

(1) Offset ≤ 5 degrees. Align the course through LTP.

(2) Offset > 5 degrees and 10 ≤ degrees. The course must cross the runway centerline extended at least 1500 feet prior to LTP (5200 feet maximum).

(3) Offset > 10 degrees and ≤ 20 degrees. The course must cross the runway centerline extended at least 3000 feet prior to LTP (5200 feet maximum). For offsets > 15 degrees, CAT C/D minimum published visibility 1 SM, minimum HAT of 300 feet.

(4) Offset > 20 to 30 degrees (CAT A/B only). The course must cross the runway centerline extended at least 4500 feet prior to the LTP (5200 feet maximum).

Note: Where offset alignment as specified above cannot be attained or the final course does not intersect the runway centerline or intersects the centerline more than 5200 feet from LTP, and an operational advantage can be achieved, the final may be aligned to lie laterally within 500 feet of the extended runway centerline at a point 3000 feet outward from LTP. For the purposes of OEA construction, use the Final End Point (FEP) as the FTP [see Order 8260.3, paragraph 2-6-2]. This option requires Flight Standards approval.

b. Circling. The optimum final course alignment is to the center of the landing area, but may be to any portion of the usable landing surface. The latest point the MAP can be located is the FEP. For the purposes of OEA construction, use the FEP as the FTP.

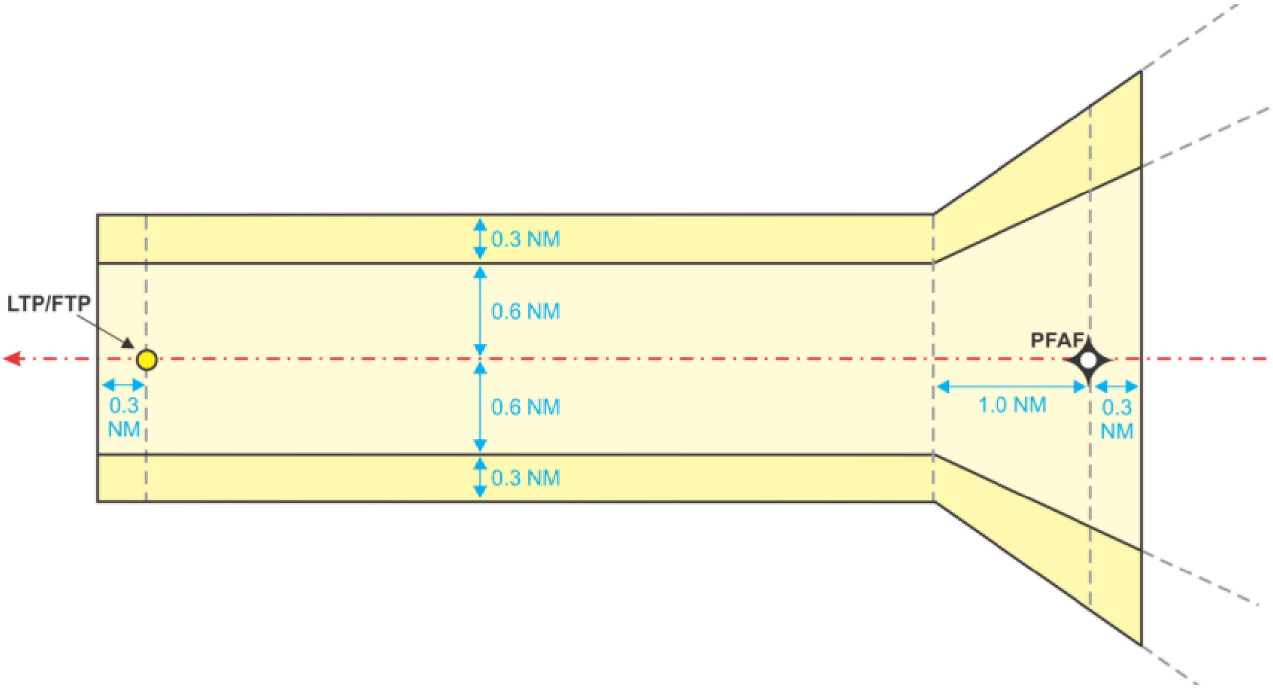
3-2-3. Area.

a. LNAV.

(1) Length. The OEA begins 0.3 NM prior to the PFAF and ends 0.3 NM past the LTP/FTP. Segment length is the distance from the PFAF location to the LTP/FTP. The maximum length is 10 NM.

(2) Width. The final segment OEA primary and secondary boundaries are coincident with the tapering intermediate segment boundaries from a point 0.3 NM prior to the PFAF to a point 1.0 NM past the PFAF. From this point to 0.3 NM past the LTP/FTP, the primary OEA boundary is ± 0.6 NM from course centerline. A 0.3 NM secondary area is located on each side of the primary area. Determine the primary and secondary area widths in this tapering area using formula 3-2-1.

Figure 3-2-1. LNAV Final Segment OEA



Formula 3-2-1. Tapering Area Width

$$^{1}/_{2}\,W_p = \frac{1.4 \times D_{taper}}{3} + 0.6$$

$$W_s = \frac{0.7 \times D_{taper}}{3} + 0.3$$

Where:
D_{taper} = along track distance (NM) from end of taper in the final segment to the point of interest

Example:

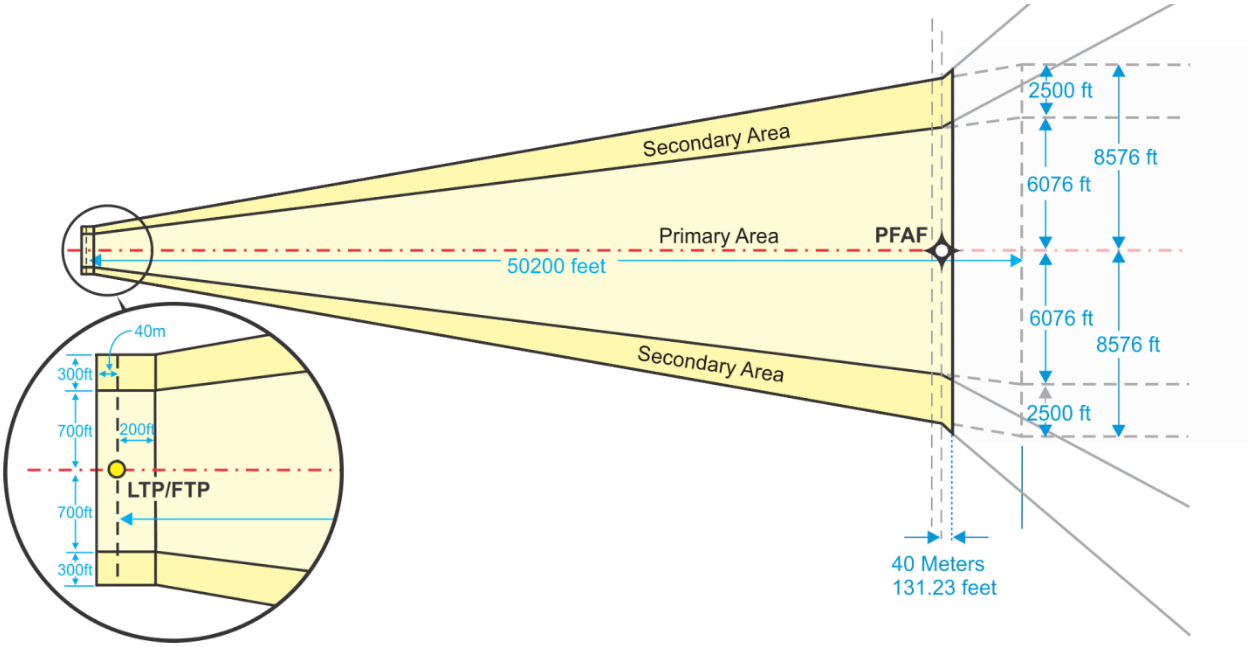
$$^{1}/_{2}\,W_p = \frac{1.4 \times 1.3}{3} + 0.6$$
$$^{1}/_{2}\,W_p \approx 1.21\text{NM}$$

$$W_s = \frac{0.7 \times 1.3}{3} + 0.3$$
$$W_s \approx 0.60\text{ NM}$$

Note: This formula may only be used when the intermediate and final legs are aligned.

b. LP.

Figure 3-2-2. LP Final OEA



(1) Length. The OEA begins 40 meters prior to the PFAF and ends 40 meters past the LTP/FTP. Segment length is the distance from the PFAF location to the LTP/FTP. The maximum length is 10 NM.

(2) Width.

(a) Primary. The perpendicular distance from the course centerline to the outer boundary of the primary area is a constant 700 feet from a point 40 meters past (inside) the LTP/FTP to a point 200 feet prior to (outside) the LTP/FTP. It expands from this point in a direction toward the PFAF. Calculate primary half width from the 200-foot point to a point 50200 feet from LTP using formula 3-2-2. The value of primary half width beyond the 50200-foot point is 6076 feet.

Formula 3-2-2. Primary Area Half Width

$$\frac{1}{2}\,\mathrm{w_p} = 0.10752 \times \mathrm{d_{LTP}} + 678.496$$

Where:
$\mathrm{d_{LTP}}$ = along track distance (feet) from LTP to point of interest

Note: 200 feet $\le \mathrm{d_{LTP}} \le$ 50200 feet

Example:

$\frac{1}{2}\,\mathrm{w_p} = 0.10752 \times 30308.58 + 678.496$
$\frac{1}{2}\,\mathrm{w_p} \approx 3937.27$ feet

(b) Secondary. The secondary width is a constant 300 feet from a point 40 meters past (inside) the LTP/FTP to a point 200 feet prior to (outside) the LTP/FTP. It expands from this point in a direction toward the PFAF. Calculate secondary width from the 200-foot point to a point 50200 feet from LTP using formula 3-2-3. The value of secondary width beyond the 50200-foot point is 2500 feet.

Formula 3-2-3. Secondary Area Width

$$\mathrm{w_s} = 0.044 \times \mathrm{d_{LTP}} + 291.2$$

Where:
$\mathrm{d_{LTP}}$ = along track distance (feet) from LTP to point of interest

Note: 200 feet $\le \mathrm{d_{LTP}} \le$ 50200 feet

Example:

$\mathrm{w_s} = 0.044 \times 30308.58 + 291.2$
$\mathrm{w_s} \approx 1624.78$ feet

3-2-4. Obstacle clearance. The minimum ROC in the primary area is 250 feet. The minimum ROC in the secondary area is 250 feet at the primary boundary, tapering uniformly to zero feet at the outer edge. Order 8260.3, chapter 3 precipitous terrain, remote altimeter, and excessive length of final adjustments apply. Calculate secondary ROC value in accordance with paragraph 1-3-1.b.

3-2-5. Minimum descent altitude. The MDA value is the sum of the controlling obstacle MSL plus the applicable ROC application from paragraph 3-2-4. Round the sum to the next higher 20-foot increment; e.g., 623 rounds to 640. The minimum HAT value is 250 feet.

Section 3-3. Lateral Navigation with Vertical Guidance (LNAV/VNAV) Final Segment

3-3-1. General. An LNAV/VNAV approach is a vertically-guided approach procedure using baro-VNAV or WAAS VNAV for the vertical guidance. Obstacle evaluation is based on the LNAV OEA dimensions, level surface OCS based on ROC, and a sloping baro-VNAV OCS. ROC adjustments for excessive length of final do not apply.

a. Vertical Path restrictions. The true vertical path provided by baro-VNAV is influenced by temperature variations; i.e., during periods of cold temperature, the effective glidepath may be lower than published and during periods of hot weather, the effective glidepath may be higher than published [see paragraph 3-3-4.a(1) and Order 8260.19 paragraph 4-6-8].

b. Other restrictions. LNAV/VNAV approach procedures at airports where Remote Altimeter Setting Source (RASS) is in use or where the final segment overlies precipitous terrain must be annotated to indicate the approach is not authorized for baro-VNAV systems.

3-3-2. Alignment. Optimum final segment alignment is with the runway centerline extended through the LTP. TF-TF turns are not allowed in the FAS. Where lowest minimums can only be achieved by offsetting the final course, it may be offset up to 15 degrees when the following conditions are met:

a. Offset ≤ 5 degrees. Align the course through LTP.

b. Offset > 5 degrees and ≤ 10 degrees. The course must cross the runway centerline extended at least 1500 feet (5200 feet maximum) prior to LTP.

c. Offset > 10 degrees and ≤ 15 degrees. The course must cross the runway centerline extended at least 3000 feet (5200 feet maximum) prior to LTP.

Note: For paragraphs 3-3-2.b and 3-3-2.c, determine the effective DA distance using formula 3-3-1. This distance can be used in formula 1-3-4 to determine the required altitude.

Formula 3-3-1. Offset Alignment Minimum Effective DA distance

$$\mathrm{d_{DAeffective}} = \frac{\mathrm{V_{KIAS}}^2 \times \tan(\mathrm{a_{offset}} \times 0.5) \times 1852}{\tan(18) \times 68625.4 \times 0.3048} + \mathrm{d_{LTP}}$$

Where:
$\mathrm{V_{KIAS}}$ = indicated final approach airspeed (knots) for applicable category
$\mathrm{a_{offset}}$ = degrees of offset between the final course and runway centerline extended
$\mathrm{d_{LTP}}$ = distance (feet) from LTP to the final course and runway centerline extended intersection

Example:

$$\mathrm{d_{DAeffective}} = \frac{165^2 \times \tan(12.5 \times 0.5) \times 1852}{\tan(18) \times 68625.4 \times 0.3048} + 4300.67$$
$$\mathrm{d_{DAeffective}} \approx 5113.16 \text{ feet}$$

3-3-3. Area. Apply paragraph 3-2-3.a.

3-3-4. Obstacle clearance surface (OCS). In the primary area, the elevation of the OCS at any point is the elevation of the OCS at the course centerline abeam it. The OCS in the secondary area is a 7:1 surface sloping upward from the edge of the primary area OCS perpendicular to the flight track [see formula 3-3-2 and figure 3-3-1].

Formula 3-3-2. Secondary Area Adjusted Obstacle Elevation

$$\mathrm{O_{adjusted}} = \mathrm{O_{MSL}} - \frac{\mathrm{OBS_Y} - {}^1\!/_2\,\mathrm{w_p}}{7}$$

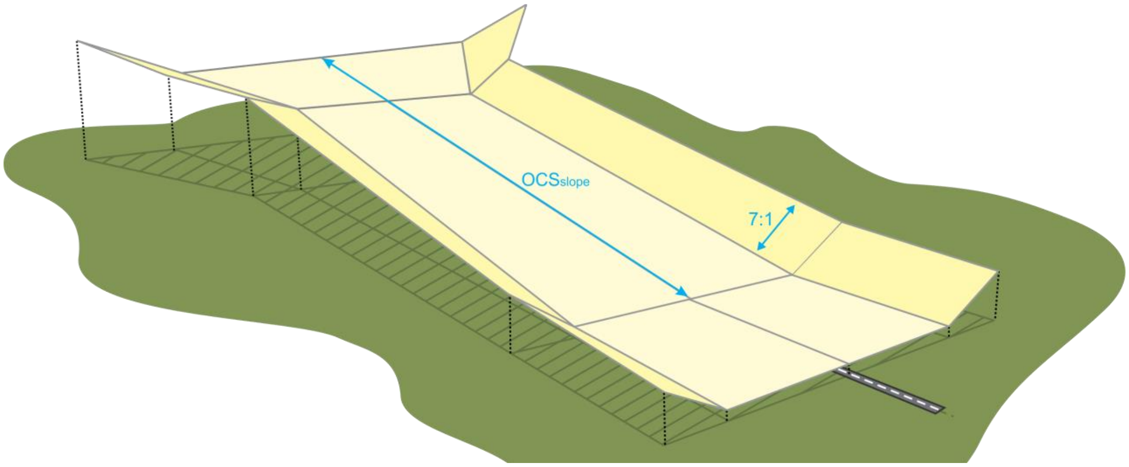
Where:
$\mathrm{OBS_Y}$ = perpendicular distance (feet) from the final approach centerline to the obstacle
${}^1\!/_2\,\mathrm{w_p}$ = perpendicular distance (feet) from final approach centerline to primary area boundary

Example:

$$\mathrm{O_{adjusted}} = \mathrm{O_{MSL}} - \frac{\mathrm{OBS_Y} - {}^1\!/_2\,\mathrm{w_p}}{7}$$
$$\mathrm{O_{adjusted}} = 3841.6 - \frac{4253.28 - 3645.67}{7}$$
$$\mathrm{O_{adjusted}} \approx 3754.80 \text{ feet}$$

a. Sloping OCS [see Order 8260.3, paragraph 2-1-4]. The primary area OCS slope varies based on critical temperatures and designed glidepath angle.

Figure 3-3-1. LNAV/VNAV Final Segment OCS



(1) Critical temperatures.

(a) Average cold temperature (ACT). In order to determine the critical low temperature, the ACT must first be established.

1. ACT based on historical temperature data.

a Source. The National Oceanic and Atmospheric Administration (NOAA), National Climatic Data Center (NCDC) is the official government source for historical temperature data.

b Reporting period. Reporting periods are established in calendar years (January through December) and must have a complete temperature record for the airport for the entire period. Ideally use the five full year period prior to the current year. If temperature data is incomplete, use the longest continuous period with complete data starting not more than six years prior to the current year. The minimum reporting period is not less than three full calendar years.

Example: The current date is in calendar year (CY) 2015. The ideal reporting period is January 1, 2010 to December 31, 2014, however temperature data is only complete up to November 2014. Since the earliest allowable year for ACT determination is 2009, the specialist chooses a reporting period from January 1, 2009 to December 31, 2013.

c Calculate the ACT as follows:

1 Find the coldest month. For each year in the reporting period, find the month with the lowest average temperature. When two or more months have the same average temperature, chose the month with the lowest single day temperature.

Example: Jan 2009: 45.4°F; Dec 2010: 44.7°F; Dec 2011: 43.4°F; Feb 2012: 42.2°F; Jan 2013: 45.0°F.

2 Find the lowest temperature in the coldest month. Find the lowest reported temperature in each of the coldest months.

Example: Jan 2009: 37°F; Dec 2010: 35°F; Dec 2011: 35°F; Feb 2012: 29°F; Jan 2013: 35°F.

3 Average the lowest temperature. When Fahrenheit values are used, convert the result to Celsius. Round the Celsius value to the next warmer whole degree; e.g., -15.00 remains -15; -14.99 becomes -14. The resultant rounded value is the ACT.

Example:

$$\mathrm{ACT} = \frac{37 + 35 + 35 + 29 + 35}{5}$$
$$\mathrm{ACT} = 34.20°\mathrm{F}$$
$$\mathrm{ACT} = (34.20 - 32)/1.8 \approx 1.22°\mathrm{C}$$
$$\mathrm{ACT} = 2°\mathrm{C}$$

2. ACT based on standard ΔISA. When historical temperature data is not available, determine ACT using a standard cold temperature deviation below airport ISA (ΔISA).

a Calculate airport ISA. Calculate the airport ISA in degrees Celsius using formula 3-3-3:

Formula 3-3-3. Airport ISA

$$\mathrm{ISA}_{\mathrm{airport}} = 15 - 0.00198 \times \mathrm{apt}_{\mathrm{elev}}$$

Example:

$$\mathrm{ISA}_{\mathrm{airport}} = 15 - 0.00198 \times 5433.8$$
$$\mathrm{ISA}_{\mathrm{airport}} \approx 4.24°\mathrm{C}$$

b Determine standard ΔISA. Select the standard ΔISA for the airport using table 3-3-1:

Table 3-3-1. Standard ΔISA Values

Airport Location	Value Below Airport ISA°C
Conus	-30°C
Alaska	-40°C
Hawaii and US territories	-20°C

c Determine the ACT. Add the standard ΔISA deviation to the airport ISA value and round the calculated value to the next warmer whole degree; e.g., -15.00 remains -15; -15.01 through -15.99 becomes -15. The resultant rounded value is the ACT.

Example:

$$\mathrm{ACT} = -30 + 4.24$$
$$\mathrm{ACT} \approx -25.76°\mathrm{C}$$
$$\mathrm{ACT} = -25°\mathrm{C}$$

(b) Determining the critical low and high temperatures. Normally, the critical low temperature limit is the calculated ACT. Where the ACT results in an effective glidepath of less than 2.5 degrees, the temperature limit is raised to achieve the required glidepath angle. The critical high temperature limit is the temp that yields an effective glidepath of 1.13 times the maximum allowed glidepath angle for the fastest designed category (not to exceed 54 degrees Celsius). Use formula 3-3-4, formula 3-3-5, and formula 3-3-6 to find both the critical low and critical high temperature.

Formula 3-3-4. Temperature Induced DA Deviation

$$\Delta\mathrm{DA_{Temp}} = e^{\frac{d_{250} \times \tan(\theta_{\mathrm{Temp}})}{r}} \times (r + \mathrm{LTP_{elev}} + \mathrm{TCH}) - (r + \mathrm{TDZE} + 250)$$

Where:
d_{250} = distance (feet) from LTP to a point 250 feet above TDZE using the designed GPA
θ_{Temp} = 2.5 for low temp deviations or 1.13 × the maximum allowable GPA of the fastest CAT for high temp deviations

Example:

$$\Delta\mathrm{DA_{Temp}} = e^{\frac{3805.69 \times \tan(2.5)}{r}} \times (r + 5321.8 + 55) - (r + 5326.3 + 250)$$
$$\Delta\mathrm{DA_{Temp}} \approx -33.30 \text{ feet}$$

Note: This calculation is accomplished twice. Once for low and once for high temperature deviations.

Formula 3-3-5. ∆ISA Adjusted

$$\Delta\mathrm{ISA_{adjusted}} = \frac{\Delta\mathrm{DA_{Temp}} \times [288 - 0.5 \times 0.00198 \times (\mathrm{TDZE} + 250)]}{250 - \Delta\mathrm{DA_{Temp}}}$$

Where:
$\Delta\mathrm{DA_{Temp}}$ = calculated maximum DA deviation if flown at the respective critical temperatures

Example:

$$\Delta\mathrm{ISA_{adjusted}} = \frac{-33.30 \times [288 - 0.5 \times 0.00198 \times (5326.3 + 250)]}{250 + 33.30}$$
$$\Delta\mathrm{ISA_{adjusted}} \approx -33.20°\mathrm{C}$$

Note 1: This calculation is accomplished twice. Once for low and once for high temperature deviations.

Note 2: $\Delta\mathrm{ISA}_{\mathrm{Low}}$ = warmer of $\Delta\mathrm{ISA}$ or $\Delta\mathrm{ISA}_{\mathrm{adjusted}}$ for low temperature deviations.

Formula 3-3-6. Adjusted ACT

$$\mathrm{ACT}_{\mathrm{adjusted}} = \mathrm{ISA}_{\mathrm{airport}} + \Delta\mathrm{ISA}_{\mathrm{adjusted}}$$

Where:
$\mathrm{ISA}_{\mathrm{airport}}$ = International Standard Atmosphere temperature °C for the airport
$\Delta\mathrm{ISA}_{\mathrm{adjusted}} =$ ISA adjustment (°C) required to achieve the respective low or high temp GPA

Example:

$\mathrm{ACT}_{\mathrm{adjusted}} = 4.24 - 33.20$
$\mathrm{ACT}_{\mathrm{adjusted}} \approx -28.96^{\circ}$

Note 1: $\mathrm{Critical}_{\mathrm{Low}}$ = warmer of ACT or $\mathrm{ACT}_{\mathrm{adjusted}}$

Note 2: $\mathrm{Critical}_{\mathrm{High}}$ = colder of 54°C or $\mathrm{ACT}_{\mathrm{adjusted}}$

(c) High temperature descent rate. Use formula 3-3-7 to determine the maximum expected descent rate at standard and critical high temperatures [see Order 8260.19, paragraph 8-7-1].

Formula 3-3-7. Descent Rate

$$\mathrm{dr} = \frac{(V_{\mathrm{KTAS}} + 10) \times 1852}{0.3048} \times \frac{60 \times \tan(\theta)}{3600}$$

Where:
V_{KTAS} = KTAS using 250 feet above TDZE and final approach KIAS for fastest CAT
$\theta =$ either designed GPA or 1.13 × maximum allowable GPA for the fastest CAT

Example:

$\mathrm{dr} = \frac{(184.10 + 10) \times 1852}{0.3048} \times \frac{60 \times \tan(3.50)}{3600}$
$\mathrm{dr} \approx 1202.23$ feet/minute

(2) OCS slope. The OCS slope and origin are determined using formula 3-3-8, formula 3-3-9, and formula 3-3-10:

Formula 3-3-8. OCS Slope

$$\mathrm{OCS_{slope}} = \frac{1}{\tan(\theta) \times \left[0.928 + 0.0038 \times (\mathrm{Critical_{Low}} - \mathrm{ISA_{airport}})\right]}$$

Where:
$\mathrm{Critical_{Low}}$ = calculated temperature (°C) that ensures no less than a 2.5° GPA
$\mathrm{ISA_{airport}}$ = International Standard Atmosphere temperature (°C) for the airport

Example:

$$\mathrm{OCS_{slope}} = \frac{1}{\tan(3) \times [0.928 + 0.0038 \times (-28.96 - 4.24)]}$$
$$\mathrm{OCS_{slope}} \approx 23.80$$

Formula 3-3-9. Sloping OCS Origin

$$\mathrm{d_{origin}} = \frac{\mathrm{TDZE} + 250 - (\mathrm{LTP_{elev}} + \mathrm{TCH})}{\tan(\theta)}$$

Example:

$$\mathrm{d_{origin}} = \frac{5326.3 + 250 - (5321.8 + 55)}{\tan(3)}$$
$$\mathrm{d_{origin}} \approx 3806.69 \text{ feet}$$

Formula 3-3-10. Distance From LTP that the Sloping OCS Application Begins

$$\mathrm{d_{OCS}} = \mathrm{d_{origin}} + \mathrm{r} \times \mathrm{OCS_{slope}} \times \ln\left(\frac{\mathrm{TDZE} + 89 + \mathrm{r}}{\mathrm{r} + \mathrm{LTP_{elev}}}\right)$$

Where:
d_{origin} = distance (feet) from LTP to OCS origin

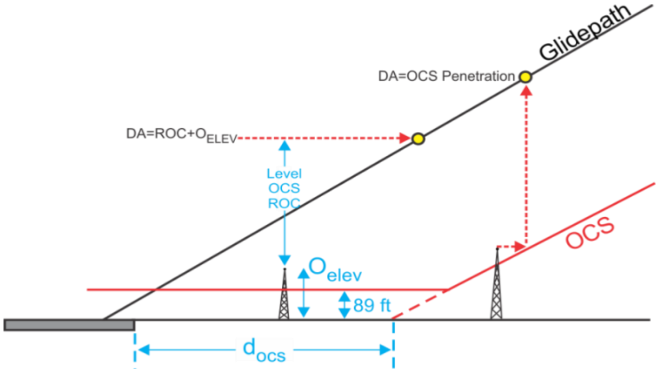
Example:

$$\mathrm{d_{OCS}} = 3806.69 + \mathrm{r} \times 23.80 \times \ln\left(\frac{5326.3 + 89 + \mathrm{r}}{\mathrm{r} + 5321.8}\right)$$
$$\mathrm{d_{OCS}} \approx 6031.42 \text{ feet}$$

(3) Obstacle accuracies/ROC adjustments. Apply the non-RNP procedure evaluation sequence using a preliminary controlling obstacle based on the sloping surface evaluation method [see Order 8260.19, paragraphs 2-11-3 and 2-11-4]. Do not apply Order 8260.3, chapter 3 precipitous terrain and/or RASS adjustments when determining preliminary DA.

(4) Sloping OCS DA adjustment [see figure 3-3-2]. The MSL elevation of the sloping OCS at any distance from LTP (beyond the OCS origin) is determined using formula 3-3-11. Where obstacles penetrate the OCS, determine the distance to the adjusted DA by entering the penetrating obstacle's actual or adjusted elevation [see formula 3-3-2] into formula 3-3-12.

Figure 3-3-2. LNAV/VNAV DA Adjustment



Formula 3-3-11. Sloping OCS Elevation

$$\mathrm{OCS_{elev}} = e^{\left(\frac{d_{LTP} - d_{origin}}{r \times OCS_{slope}}\right)} \times (r + \mathrm{LTP_{elev}}) - r$$

Where:
d_{LTP} = along track distance (feet) from LTP to point of interest
d_{origin} = distance (feet) from LTP to OCS origin

Example:

$$\mathrm{OCS_{elev}} = e^{\left(\frac{19747.38 - 3911.63}{r \times 23.42}\right)} \times (r + 1125.4) - r$$
$$\mathrm{OCS_{elev}} \approx 1801.61 \text{ feet}$$

Formula 3-3-12. Preliminary DA Distance Based On Sloping OCS Penetration

$$d_{DA} = (r + \mathrm{LTP_{elev}}) \times \mathrm{OCS_{slope}} \times \ln\left(\frac{r + O_{elev}}{r + \mathrm{LTP_{elev}}}\right) + d_{origin}$$

Where:
O_{elev} = obstacle actual or $O_{adjusted}$ elevation (feet)
d_{origin} = distance (feet) from LTP to OCS origin

Example:

$$d_{DA} = (r + 1125.4) \times 23.42 \times \ln\left(\frac{r + 1263.7}{r + 1125.4}\right) + 3911.63$$
$$d_{DA} \approx 7150.61 \text{ feet}$$

b. Level OCS [see Order 8260.3, paragraph 2-1-3]. A level OCS is applied from 0.3 NM past the LTP/FTP to the point of intersection with the sloping OCS. If an obstacle is in the secondary area (transitional surface), adjust the height of the obstacle using formula 3-3-2, then evaluate it at the adjusted height as if it is in the primary area.

(1) Obstacle clearance. Apply the applicable ROC from table 3-3-2 to the obstacle with the highest effective height to determine the preliminary DA.

(2) Obstacle accuracies/ROC adjustments. Apply the non-RNP procedure evaluation sequence using a preliminary controlling obstacle based on the level surface evaluation method [see Order 8260.19, paragraphs 2-11-3 and 2-11-4]. Do not apply Order 8260.3, chapter 3 precipitous terrain and/or RASS adjustments when determining preliminary DA.

Table 3-3-2. Level OCS ROC by Approach Category

Aircraft Category	ROC
A	131
B	142
C	150
D/E	161

3-3-5. Decision altitude (DA). The minimum HAT is 250 feet (Order 8260.3, table 3-2-2 does not apply). The DA based on the final segment evaluation is the highest of the following; the DA based on the minimum HAT, the DA based on the minimum offset DA distance [see paragraph 3-3-2], the DA based on evaluation of the level OCS, or the DA based on evaluation of the sloping OCS. APV precipitous terrain and RASS adjustments apply. The DA based on the evaluation of the sloping OCS is not required to be higher than the MDA of an LNAV procedure established on the same chart. Order 8260.19, paragraph 8-7-1.b applies, except document the controlling obstacle and ROC based on whichever evaluation (level or sloping OCS) results in the highest DA.

3-3-6. Raising GPA or adjusting TCH for penetrating obstructions. The GPA and/or TCH may be raised to eliminate or reduce final segment OCS penetrations.

Section 3-4. LPV/GLS Final

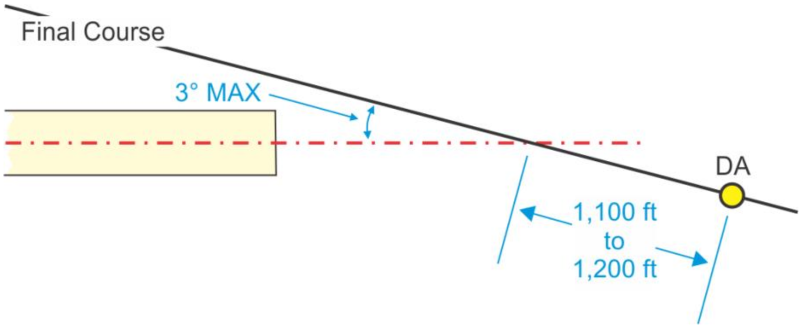
3-4-1. General. The OEA and associated OCSs are applicable to LPV and GLS final approach segments.

a. Exceptions. Paragraph 1-3-1 vertical path requirements apply except for procedures with HAT less than 250, the maximum GPA is 3.5 degrees and the maximum TCH is 60 feet regardless of height group.

b. Restrictions. For GLS procedures, final track intercept must occur within 20 NM of the airport.

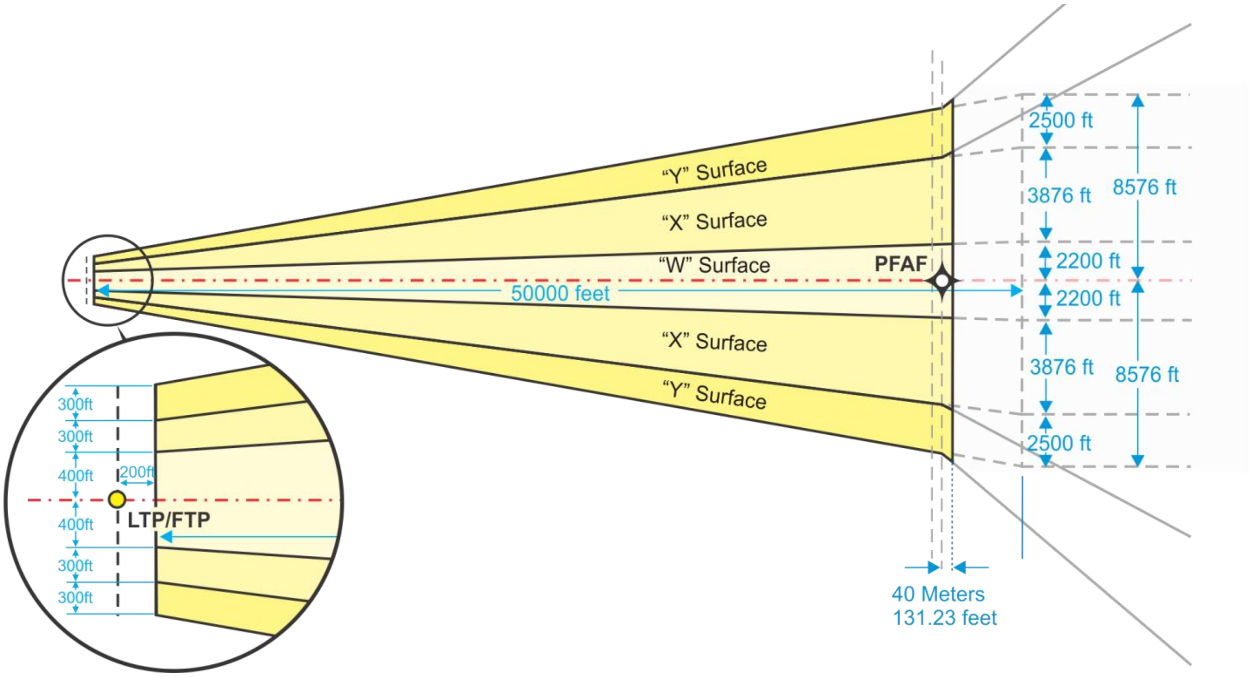
3-4-2. Alignment. The final course is normally aligned with the runway centerline (RCL) extended through the LTP (± 5 feet). Where an operations requirement necessitates an offset course, the FAC must intersect the RCL extended 1100 to 1200 feet inside the DA point and the offset must not exceed three degrees measured from the point of intersection [see figure 3-4-1]. Where the course is offset, the minimum HAT is 250. TF-TF turns are not allowed in the FAS.

Figure 3-4-1. Offset Final Alignment



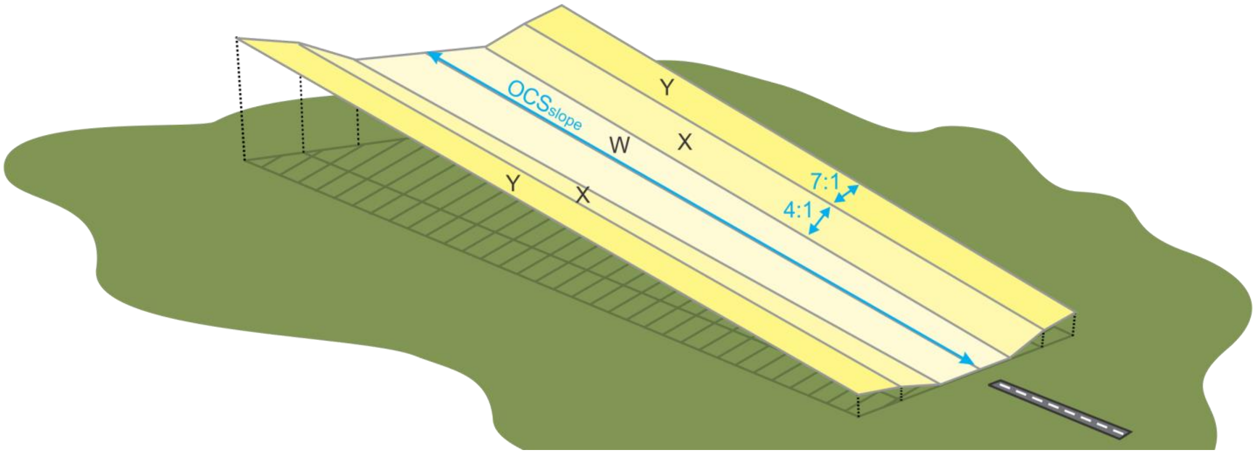
3-4-3. Area. The OEA originates 200 feet from LTP and extends to a point 40 meters beyond the PFAF. It is centered on the final approach course and expands uniformly from its origin to a point 50000 feet from the origin. Where the PFAF must be located more than 50200 feet from LTP, the OEA continues linearly (boundaries parallel to course centerline) beyond that point [see figure 3-4-2].

Figure 3-4-2. LPV/GLS Final OEA



3-4-4. Obstacle clearance surface. The primary area OCS consists of the W and X surfaces. The Y surface is an early missed approach transitional surface. The W surface slopes longitudinally along the final approach track, and is level perpendicular to track. The X and Y surfaces slope upward from the edge of the W surface perpendicular to the final approach track [see figure 3-4-3]. Obstacles located in the X and Y surfaces are adjusted in height to account for perpendicular surface rise and evaluated under the W surface.

Figure 3-4-3. LPV/GLS Final OCSs



a. OCS slope. Determine the OCS slope associated with a specific glidepath angle using formula 3-4-1.

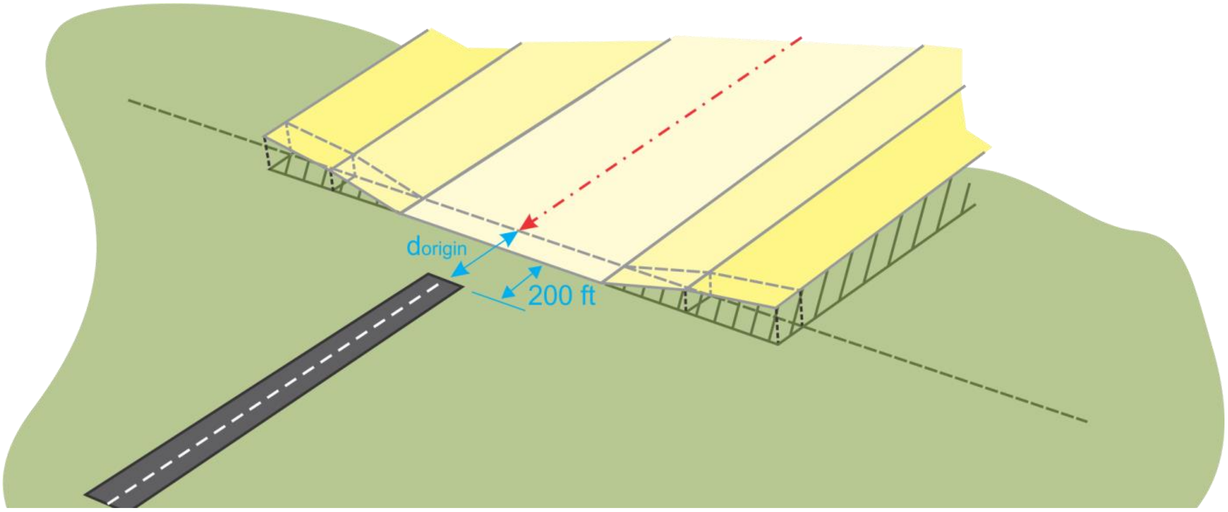
Formula 3-4-1. OCS Slope

$$\text{OCS}_{\text{slope}} = \frac{102}{\theta}$$

Example:

$$\text{OCS}_{\text{slope}} = \frac{102}{3.1}$$
$$\text{OCS}_{\text{slope}} \approx 32.90$$

Figure 3-4-4. OCS Slope Origin



b. OCS origin. All OCS surfaces begin at LTP elevation at a point 200 feet from LTP [see figure 3-4-4] measured along course centerline and extends toward the PFAF. The longitudinal (along-track) rising W surface slope begins at a calculated distance from the LTP [see formula 3-4-2].

Formula 3-4-2. OCS Origin

$$d_{\text{origin}} = \text{greater of } 200 \text{ or } 1154 - \frac{\text{TCH}}{\tan(\theta)}$$

Example:

$$d_{\text{origin}} = 1154 - \frac{55}{\tan(3.1)}$$
$$d_{\text{origin}} \approx 138.45 \text{ feet}$$
$$d_{\text{origin}} = 200 \text{ feet}$$

c. Obstacle effective elevation ($\mathrm{O_{EE}}$). Because the earth curves away from the OCS as distance from course centerline increases, the MSL elevation of an obstacle is reduced to account for this. Use formula 3-4-3 to calculate the $\mathrm{O_{EE}}$.

Formula 3-4-3. Obstacle Effective Elevation

$$\mathrm{O_{EE}} = \mathrm{O_{MSL}} - \left[(\mathrm{r} + \mathrm{LTP_{elev}}) \times \left(\cos\left[\frac{\mathrm{OBS_Y} \times 180}{\mathrm{r} \times \pi}\right]^{-1} - 1\right) + \mathrm{Q}\right]$$

Where:
$\mathrm{OBS_Y}$ = perpendicular distance (feet) from the course centerline to the obstacle
Q = obstacle adjustment (feet) for X or Y surface rise. Zero (0) if in the W surface.

Example:

$$\mathrm{O_{EE}} = 2768.9 - \left[(\mathrm{r} + 1125.4) \times \left(\cos\left[\frac{1432.5 \times 180}{\mathrm{r} \times \pi}\right]^{-1} - 1\right) + 192.90\right]$$
$$\mathrm{O_{EE}} \approx 2575.95 \text{ feet}$$

d. W OCS.

(1) Width. The perpendicular distance from course centerline to the boundary is 400 feet at the origin, and expands uniformly to 2200 feet at a point 50200 feet from LTP/FTP. Calculate the OCS half-width using formula 3-4-4.

Formula 3-4-4. W OCS Half Width

$$\mathrm{W_{boundary}} = 0.036 \times \mathrm{d_{LTP}} + 392.8$$

Where:
$\mathrm{d_{LTP}}$ = along track distance (feet) from LTP to point of interest

Note: 200 feet ≤ $\mathrm{d_{LTP}}$ ≤ 50200 feet

Example:

$$\mathrm{W_{boundary}} = 0.036 \times 5462.03 + 392.8$$
$$\mathrm{W_{boundary}} \approx 589.43 \text{ feet}$$

(2) OCS elevation. The LPV/GLS glidepath is considered to be a straight line in space extending from TCH. To protect the straight-line glidepath, the OCS is a flat plane (does not follow earth curvature). The elevation of the W OCS at any point is the elevation of the OCS at the course centerline abeam it. Calculate the W OCS angle using formula 3-4-5 and elevation at any distance beyond the OCS origin using formula 3-4-6.

Formula 3-4-5. OCS Angle

$$\mathrm{OCS_{angle}} = \mathrm{atan}\left(\frac{\theta}{102}\right)$$

Example:

$\mathrm{OCS_{angle}} = \mathrm{atan}\left(\frac{3.1}{102}\right)$
$\mathrm{OCS_{angle}} \approx 1.74^{\circ}$

Formula 3-4-6. W OCS Elevation

$$\mathrm{OCS_{elev}} = \frac{(\mathrm{r} + \mathrm{LTP_{elev}}) \times \cos(\mathrm{OCS_{angle}})}{\cos\left[\frac{(\mathrm{d_{LTP}} - \mathrm{d_{origin}}) \times 180}{\mathrm{r} \times \pi} + \mathrm{OCS_{angle}}\right]} - \mathrm{r}$$

Where:
$\mathrm{OCS_{angle}}$ = calculated angle of the obstacle clearance surface
$\mathrm{d_{LTP}}$ = along track distance (feet) from LTP to point of interest
$\mathrm{d_{origin}}$ = distance (feet) from LTP to OCS origin

Example:

$\mathrm{OCS_{elev}} = \frac{(\mathrm{r} + 1125.4) \times \cos(1.74)}{\cos\left[\frac{(5280 - 200) \times 180}{\mathrm{r} \times \pi} + 1.74\right]} - \mathrm{r}$
$\mathrm{OCS_{elev}} \approx 1280.35$ feet

(3) OCS evaluation. Compare the W OCS elevation abeam the obstacle location with the $\mathrm{O_{EE}}$. Lowest minimums are achieved when the W surface is clear. To eliminate or avoid a penetration, take one or more of the following actions:

(a) Remove or adjust the obstruction location and/or height.

(b) Displace the RWT.

(c) Raise GPA within the categorical limits.

(d) Adjust DA.

(e) Raise TCH.

e. X OCS.

(1) Width. The perpendicular distance from the course centerline to the outer boundary of the X OCS is 700 feet at the origin and expands uniformly to 6076 feet at a point 50200 feet from LTP/FTP. Calculate the perpendicular distance from the course centerline to the X surface boundary using formula 3-4-7.

Formula 3-4-7. Perpendicular Distance to X Boundary

$$X_{\text{boundary}} = 0.10752 \times d_{\text{LTP}} + 678.496$$

Where:
d_{LTP} = along track distance (feet) from LTP to point of interest

Note: 200 feet ≤ d_{LTP} ≤ 50200 feet

Example:

$X_{\text{boundary}} = 0.10752 \times 5462.03 + 678.496$
$X_{\text{boundary}} \approx 1265.77$ feet

(2) Obstacle adjustment. The X OCS begins at the height of the W surface and rises at a slope of 4:1 in a direction perpendicular to the final approach course. The MSL elevation of an obstacle in the X surface is reduced by the amount of surface rise. Use formula 3-4-8 to determine the obstacle height adjustment (Q) for use in formula 3-4-3. Evaluate the obstacle in accordance with paragraph 3-4-4.d(3).

Formula 3-4-8. X OCS Obstacle Adjustment

$$Q = \frac{OBS_Y - W_{\text{boundary}}}{4}$$

Where:
OBS_Y = perpendicular distance (feet) from the course centerline to the obstacle
W_{boundary} = perpendicular distance (feet) from the course centerline to the W surface boundary

Example:

$Q = \frac{1265.77 - 589.43}{4}$
$Q \approx 169.09$ feet

f. Y OCS.

(1) Width. The perpendicular distance from the course centerline to the outer boundary of the Y OCS is 1000 feet at the origin and expands uniformly to 8576 feet at a point 50200 feet from LTP/FTP. Calculate the perpendicular distance from the course centerline to the Y surface boundary using formula 3-4-9.

Formula 3-4-9. Perpendicular Distance to Y Boundary

$$\mathrm{Y_{boundary}} = 0.15152 \times \mathrm{d_{LTP}} + 969.696$$

Where:
$\mathrm{d_{LTP}}$ = along track distance (feet) from LTP to point of interest

Note: 200 feet ≤ $\mathrm{d_{LTP}}$ ≤ 50200 feet

Example:

$\mathrm{Y_{boundary}}$=$0.15152 \times 5462.03 + 969.696$
$\mathrm{Y_{boundary}} \approx 1797.30$ feet

(2) Obstacle adjustment. The Y OCS begins at the height of the X surface and rises at a slope of 7:1 in a direction perpendicular to the final approach course. The MSL elevation of an obstacle in the Y surface is reduced by the amount of X and Y surface rise. Use formula 3-4-10 to determine the obstacle height adjustment (Q) for use in formula 3-4-3. Evaluate the obstacle in accordance with paragraph 3-4-4.d(3).

Formula 3-4-10. Y OCS Obstacle Adjustment

$$\mathrm{Q} = \frac{\mathrm{X_{boundary}} - \mathrm{W_{boundary}}}{4} + \frac{\mathrm{OBS_Y} - \mathrm{X_{boundary}}}{7}$$

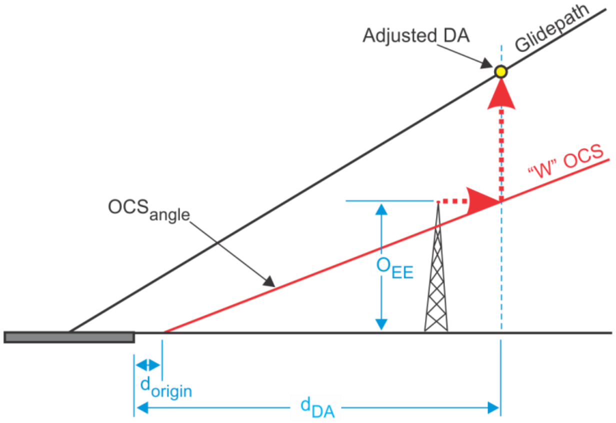
Where:
$\mathrm{X_{boundary}}$ = perpendicular distance (feet) from the course centerline to the X surface boundary
$\mathrm{W_{boundary}}$ = perpendicular distance (feet) from the course centerline to the W surface boundary
$\mathrm{OBS_Y}$ = perpendicular distance (feet) from the course centerline to the obstacle

Example:

$$\mathrm{Q} = \frac{1265.77 - 589.43}{4} + \frac{1432.5 - 1265.77}{7}$$
$\mathrm{Q} \approx 192.90$ feet

3-4-5. Decision altitude (DA). When the final OCS is clear, the minimum HAT is 200 feet. When the final OCS is penetrated or when the final course is offset, the minimum HAT is 250. The DA is the highest of the following; the DA based on the applicable minimum HAT; the DA based on GPA [see Order 8260.3, table 3-2-2]; the DA based on evaluation of the final segment OCS. Calculate the adjusted DA distance for OCS penetrations using formula 3-4-11 for input into formula 3-4-12.Calculate the published DA distance from the LTP/FTP using formula 3-4-13. The published DA is not required to be higher than the MDA of an LNAV procedure established on the same chart.

Figure 3-4-5. LPV/GLS DA Adjustment



Formula 3-4-11. Adjusted DA distance

$$d_{DA} = \frac{r \times \pi}{180} \times \left(90 - OCS_{angle} - \operatorname{asin}\left[\frac{\cos\left(OCS_{angle}\right) \times (r + LTP_{elev})}{r + O_{EE}}\right]\right) + d_{origin}$$

Where:
OCS_{angle} = calculated angle of the obstacle clearance surface
d_{origin} = distance (feet) from LTP to OCS origin
O_{EE} = calculated obstacle effective elevation (feet)

Example:

$$d_{DA} = \frac{r \times \pi}{180} \times \left(90 - 1.74 - \operatorname{asin}\left[\frac{\cos(1.74) \times (r + 1125.4)}{r + 1271.5}\right]\right) + 200$$

$d_{DA} \approx 4991.01$ feet

Formula 3-4-12. Altitude on LPV/GLS Glidepath

$$Z_{LPV} = \frac{(r + alt_b) \times \cos(\theta)}{\cos\left(\frac{d \times 180}{r \times \pi} + \theta\right)} - r$$

Where:
alt_b = beginning altitude (feet) [e.g., LTP elevation + TCH]
d = distance (feet) between the points of interest

Example:

$$Z_{LPV} = \frac{(r + (1125.4 + 55)) \times \cos(3.1)}{\cos\left(\frac{42041.91 \times 180}{r \times \pi} + 3.1\right)} - r$$

$Z_{LPV} \approx 3500$ feet

Formula 3-4-13. Distance on LPV/GLS Glidepath

$$\mathrm{d_{LPV}} = \frac{\mathrm{r} \times \pi}{180} \times \left(90 - \theta - \mathrm{asin}\left[\frac{\cos(\theta) \times (\mathrm{r} + \mathrm{alt_b})}{\mathrm{r} + \mathrm{alt_e}}\right]\right)$$

Where:
$\mathrm{alt_b}$ = beginning altitude (MSL)
$\mathrm{alt_e}$ = ending altitude (MSL)

Example:

$$\mathrm{d_{LPV}} = \frac{\mathrm{r} \times \pi}{180} \times \left(90 - 3.1 - \mathrm{asin}\left[\frac{\cos(3.1) \times (\mathrm{r} + 1125.4 + 55)}{\mathrm{r} + 3500}\right]\right)$$

$\mathrm{d_{LPV}} \approx 42041.91$ feet

3-4-6. Raising GPA for OCS penetrations. Raising the GPA may eliminate OCS penetrations. To determine the revised minimum GPA, use formula 3-4-14.

Formula 3-4-14. GPA Adjustment

$$\mathrm{SRD} = \sqrt{(\mathrm{r} + \mathrm{O_{EE}})^2 + (\mathrm{r} + \mathrm{LTP_{elev}})^2 - 2 \times (\mathrm{r} + \mathrm{O_{EE}}) \times (\mathrm{r} + \mathrm{LTP_{elev}}) \times \cos\left[\frac{(\mathrm{d_{LTP}} - \mathrm{d_{origin}}) \times 180}{\mathrm{r} \times \pi}\right]}$$

$$\theta_{\mathrm{adjusted}} = \tan\left[\mathrm{acos}\left(\frac{\mathrm{SRD}^2 + (\mathrm{r} + \mathrm{LTP_{elev}})^2 - (\mathrm{r} + \mathrm{O_{EE}})^2}{2 \times \mathrm{SRD} \times (\mathrm{r} + \mathrm{LTP_{elev}})}\right) - 90\right] \times 102$$

Where:
$\mathrm{O_{EE}}$ = calculated obstacle effective elevation (feet)
$\mathrm{d_{LTP}}$ = along track distance (feet) from LTP to penetrating obstacle
$\mathrm{d_{origin}}$ = distance (feet) from LTP to OCS origin

Example:

$$\mathrm{SRD} = \sqrt{(\mathrm{r} + 1274.5)^2 + (\mathrm{r} + 1125.4)^2 - 2 \times (\mathrm{r} + 1274.5) \times (\mathrm{r} + 1125.4) \times \cos\left[\frac{(3992.7 - 200) \times 180}{\mathrm{r} \times \pi}\right]}$$

$SRD \approx 3795.85$ feet

$$\theta_{\mathrm{adjusted}} = \tan\left[\mathrm{acos}\left(\frac{3795.85^2 + (\mathrm{r} + 1125.4)^2 - (\mathrm{r} + 1274.5)^2}{2 \times 3795.85 \times (\mathrm{r} + 1125.4)}\right) - 90\right] \times 102$$

$\theta_{\mathrm{adjusted}} \approx 4.00^{\circ}$

3-4-7. Adjusting TCH to reduce/eliminate OCS penetrations. This paragraph is applicable only where the OCS origin is greater than 200 feet from the LTP. Adjusting TCH is the equivalent to relocating the glide slope antenna in ILS criteria. The goal is to move the OCS origin toward the LTP/FTP (no closer than 200 feet) to raise the OCS at the obstacle location. To determine the maximum W surface vertical relief that can be achieved by adjusting TCH, apply formula 3-4-15. If the result is greater than the penetration, you may determine the amount to increase TCH by applying formula 3-4-16. If this option is selected, re-evaluate the final segment using the revised TCH value.

Formula 3-4-15. Vertical Relief

$$\mathrm{Z_{TCH}} = \frac{\mathrm{d_{origin}} - 200}{\mathrm{OCS_{slope}}}$$

Where:
$\mathrm{d_{origin}}$ = distance (feet) from LTP to OCS origin

Example:

$$\mathrm{Z_{TCH}} = \frac{390.75 - 200}{34}$$
$\mathrm{Z_{TCH}} \approx 5.61$ feet

Formula 3-4-16. TCH Adjustment

$$\Delta\mathrm{TCH} = \tan(\theta) \times \mathrm{OCS_{slope}} \times \mathrm{p}$$

Where:
p = penetration (feet) of the final OCS

Example:

$\Delta\mathrm{TCH} = \tan(3) \times 34 \times 4.04$
$\Delta\mathrm{TCH} \approx 7.20$ feet

Section 3-5. Missed Approach General

3-5-1. General. Segment XTT is RNP APCH, missed approach flight phase. *Exception: For LNAV and LP, apply ATT based on the applicable final flight phase at the beginning of section 1 (line C-D).* Secondary areas apply where specified. Line C-D location and altitude is defined using the base MDA/DA. The base MDA/DA is the lowest value resulting from subtracting any RASS adjustments (primary and secondary setting source) from the lowest final segment MDA/DA (rounded to a publishable value).

Note: Due to rounding, it is sometimes possible for the MDA/DA minus RASS adjustment to be lower than the MDA/DA based on local altimeter.

3-5-2. Course-to-altitude (CA) leg. A CA leg is used as the first leg of an RNAV missed approach and must be followed by a DF leg. The CA leg must specify a course and altitude. The specified CA course must be an extension of the FAC. The specified CA altitude is determined as follows:

a. When the first fix after the coded MAP is on the FAC extended, the specified CA altitude is the DA, MDA, or 400 ft above the airport elevation whichever is lowest.

b. When a turning missed approach is based on a “climb-to” altitude before turning, the specified CA altitude is the charted “climb-to” altitude.

3-5-3. MA climb gradient termination. Where the default OCS slope is penetrated and the lowest HAT is required, a greater than standard CG may be required to clear the penetrating obstruction. Calculate the climb gradient termination altitude using formula 3-5-1 for input into the climb gradient formula 1-3-7. For establishing a RASS-based climb-to-altitude, see Order 8260.3, chapter 3. If a RASS adjustment is applicable, apply the CG associated with the lowest MDA/DA.

Formula 3-5-1. MA CG Termination Altitude

$$\mathrm{CG_{term}} = 48 \times \mathrm{D_0} + \mathrm{O_{MSL}} - \frac{\mathrm{d_{primary}}}{12} + \mathrm{AC_{start}} - \mathrm{OCS_{start}}$$

Where:
$\mathrm{D_0}$ = dist. (NM) from $\mathrm{AC_{start}}$ to primary obs. or point from which $\mathrm{d_{primary}}$ measured
$\mathrm{d_{primary}}$ = dist. (feet) from edge of primary area to obs. Zero (0) if not in secondary
$\mathrm{AC_{start}}$ = starting altitude (feet) of the aircraft (e.g., SOC)
$\mathrm{OCS_{start}}$ = starting altitude (feet)of the OCS

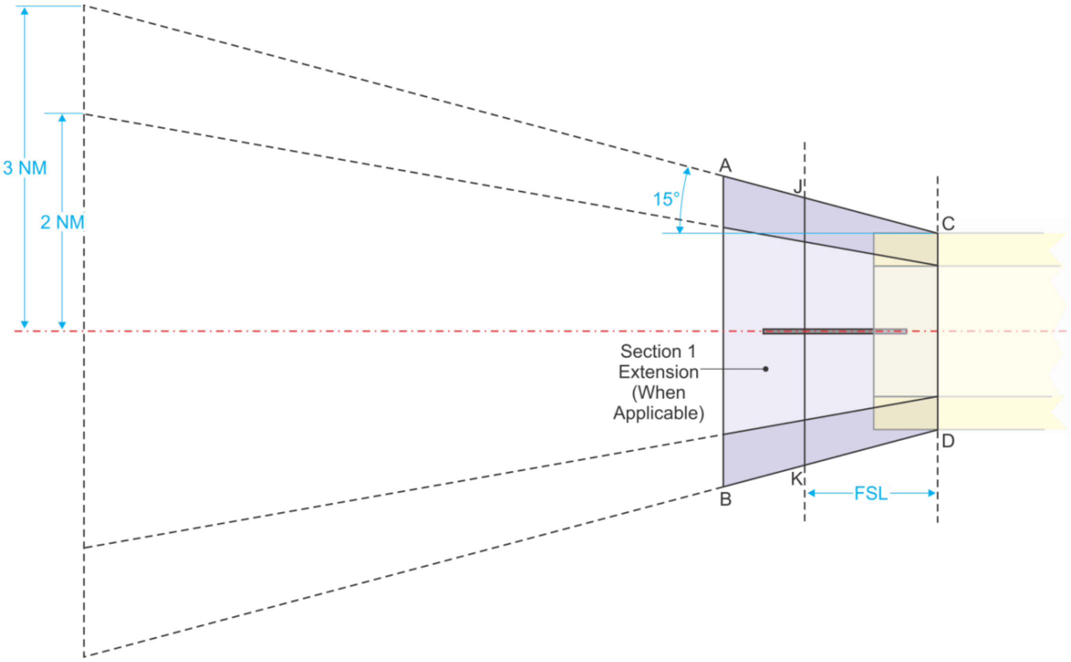
Example:

$$\mathrm{CG_{term}} = 48 \times 3.95 + 2147.41 - \frac{537.41}{12} + 1442 - 1217.21$$
$$\mathrm{CG_{term}} \approx 2517.02 \text{ feet}$$

Section 3-6. Missed Approach Section 1

3-6-1. Non-vertically guided. Section 1 begins final segment ATT prior to the MAP (line C-D) along a continuation of the final course and extends to the SOC (line J-K) or the point where the aircraft is projected to cross 400 feet above airport elevation (line A-B); whichever is the greater distance from the MAP [see figure 3-6-1 and figure 3-6-2].

Figure 3-6-1. LNAV Missed Approach Section 1 Area



a. Length.

(1) Flat surface length (FSL). The section 1 flat surface begins either 0.3 NM for LNAV or 40 meters for LP prior to the MAP and extends the FSL distance to line J-K. Calculate the FSL using formula 3-6-1.

Formula 3-6-1. LNAV/LP Flat Surface Length

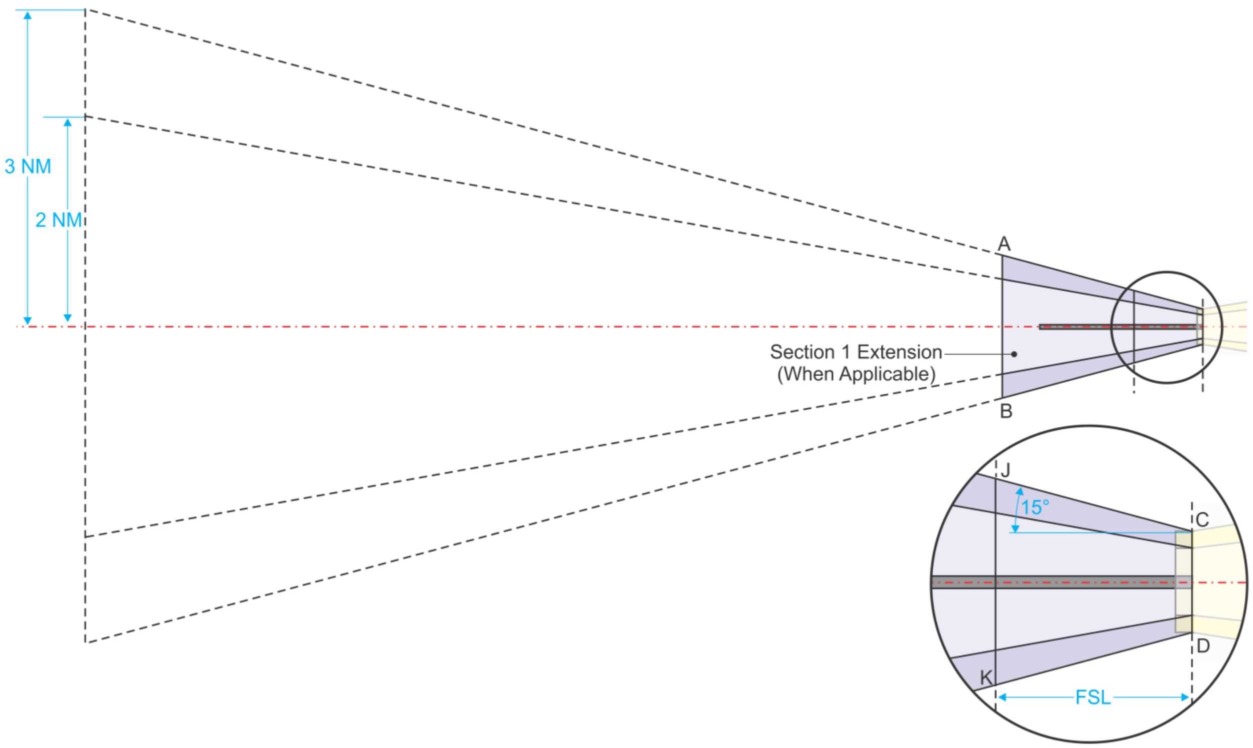
$$\mathrm{FSL} = \frac{(\mathrm{V_{KTAS}} + 10) \times 12}{3600} + 2 \times \mathrm{ATT_{NM}}$$

Where:
V_{KTAS} = calculated KTAS using MDA and final approach KIAS
ATT_{NM} = applicable final segment ATT in NM

Example:

$$\mathrm{FSL} = \frac{(171.91 + 10) \times 12}{3600} + 2 \times 0.3$$
$$\mathrm{FSL} \approx 1.21\ \mathrm{NM}$$

Figure 3-6-2. LP Missed Approach Section 1 Area



(2) Section 1 extension.

(a) Base MDA ≥ 400 feet above airport elevation. No extension required. Line A-B is coincident with line J-K.

(b) Base MDA < 400 feet above airport elevation. Calculate the section 1 extension using formula 3-6-2 and locate line A-B at this distance beyond line J-K.

Formula 3-6-2. Section 1 Extension

$$\mathrm{S1_{extension}} = \frac{\mathrm{apt_{elev}} + 400 - \mathrm{alt_b}}{\mathrm{CG}}$$

Where:
$\mathrm{alt_b}$ = either the Base MDA for LNAV/LP or Base DA for LNAV/VNAV

Example:

$$\mathrm{S1_{extension}} = \frac{3152.6 + 400 - 3471}{210}$$
$$\mathrm{S1_{extension}} \approx 0.39\ \mathrm{NM}$$

b. Width. The secondary area outer boundary starts at the outer edge of the final segment secondary area at line C-D and splays at 15-degrees relative to the missed approach course until reaching 3 NM width from centerline. The primary area outer boundary starts at the edge of the primary area at line C-D and splays at the angle required to reach a width of 2 NM from centerline adjacent to the point where secondary outer boundary reaches 3 NM width. Calculate the distance from course centerline to the primary and secondary boundary of the OEA at any distance from line C-D using formula 3-6-3 or formula 3-6-4, depending on the final type.

Formula 3-6-3. LNAV and LNAV/VNAV MA Primary and Secondary Boundary

$$\mathrm{MAS_{Yprimary}} = \frac{\mathrm{D_{splay}} \times \tan(15) \times 1.4}{2.1} + 0.6$$
$$\mathrm{MAS_{Ysecondary}} = \mathrm{D_{splay}} \times \tan(15) + 0.9$$

Where:
$\mathrm{D_{splay}}$ = along track distance (NM) from beginning of MA splay to point of interest

Example:

$$\mathrm{MAS_{Yprimary}} = \frac{6.04 \times \tan(15) \times 1.4}{2.1} + 0.6$$
$$\mathrm{MAS_{Yprimary}} \approx 1.68\ \mathrm{NM}$$

$$\mathrm{MAS_{Ysecondary}} = 6.04 \times \tan(15) + 0.9$$
$$\mathrm{MAS_{Ysecondary}} \approx 2.52\ \mathrm{NM}$$

Formula 3-6-4. LP MA Primary and Secondary Boundary

$$\mathrm{MAS_{Yprimary}} = \frac{\mathrm{D_{splay}} \times \tan(15) \times (2 - {}^{1}\!/_{2}\,\mathrm{W_p})}{3 - \mathrm{W_s} - {}^{1}\!/_{2}\,\mathrm{W_p}} + {}^{1}\!/_{2}\,\mathrm{W_p}$$
$$\mathrm{MAS_{Ysecondary}} = \mathrm{D_{splay}} \times \tan(15) + \mathrm{W_s} + {}^{1}\!/_{2}\,\mathrm{W_p}$$

Where:
$\mathrm{D_{splay}}$ = along track distance (NM) from beginning of MA splay to point of interest
${}^{1}\!/_{2}\,\mathrm{W_p}$ = perpendicular dist. (NM) from FAS centerline to FAS primary boundary
$\mathrm{W_s}$ = final secondary area width (NM)

Example:

$$\mathrm{MAS_{Yprimary}} = \frac{6.04 \times \tan(15) \times (2 - 0.20)}{3 - 0.09 - 0.20} + 0.20$$
$$\mathrm{MAS_{Yprimary}} \approx 1.27\ \mathrm{NM}$$

$$\mathrm{MAS_{Ysecondary}} = 6.04 \times \tan(15) + 0.09 + 0.20$$
$$\mathrm{MAS_{Ysecondary}} \approx 1.91\ \mathrm{NM}$$

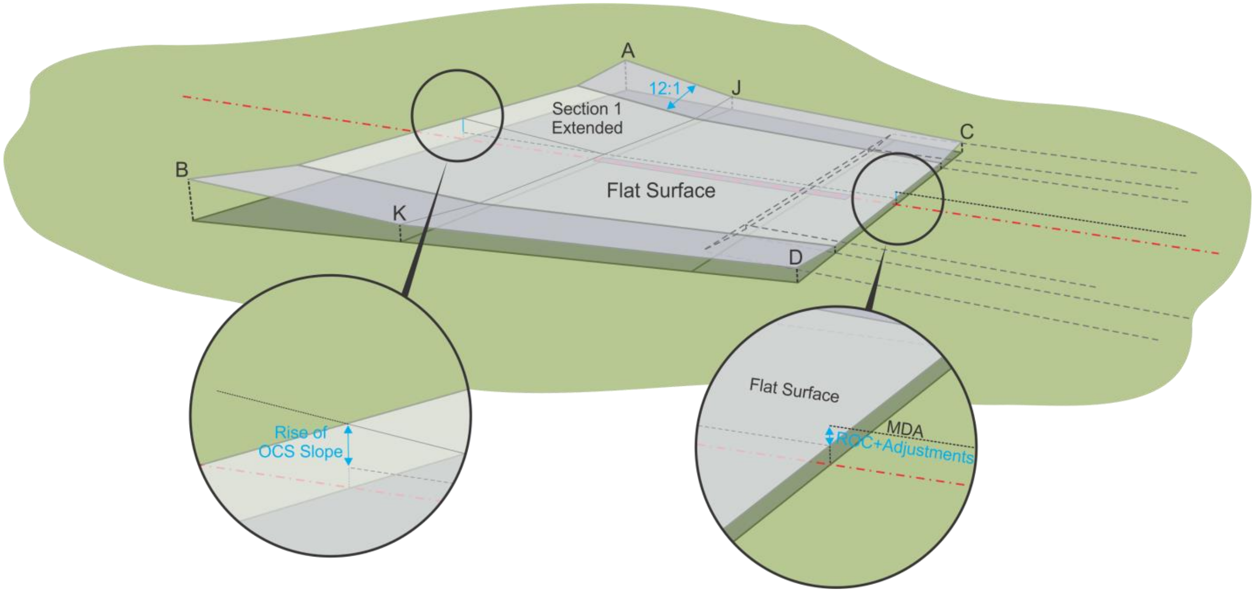
c. Obstacle clearance. Section 1 OCS consists of a flat surface and sloping surface extension [see figure 3-6-3]. The MSL height at the start of the missed approach surface (HMAS) is equal to the base MDA less 100 feet plus any precipitous terrain, and excessive length of final adjustments. When obstacles are mitigated by one of the following, re-evaluate the missed approach segment.

(1) Flat surface. The missed approach primary surface remains flat from line C-D to line J-K. Obstacles must not penetrate the flat surface. Where obstacles penetrate the flat surface, raise the MDA by the amount of penetration.

(2) Section 1 extension. From line J-K to line A-B, a sloping OCS surface rises along the missed approach course centerline at a slope ratio commensurate with the obstructions [see paragraph 1-3-1.g(1)]. Where obstacles penetrate this OCS, either adjust the MDA by the amount of penetration or apply a climb gradient to clear the obstacle [see paragraph 1-3-1.g(2)].

(3) Secondary areas apply from the edge of both the flat surface and section 1 extended with a 12:1 slope measured perpendicular to centerline.

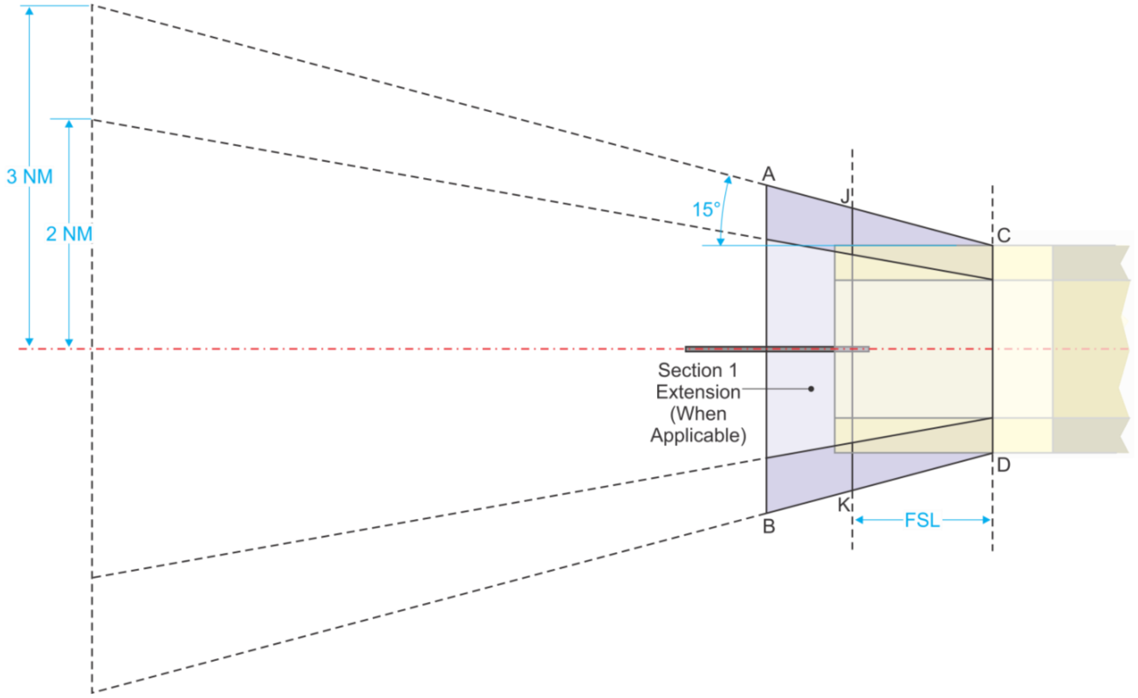
Figure 3-6-3. LNAV Missed Approach Section 1 OCS



3-6-2. LNAV/VNAV. The same criteria specified for non-vertically guided procedures apply with the following exceptions [see figure 3-6-4].

a. Line C-D is at the base DA point.

Figure 3-6-4. LNAV/VNAV Missed Approach Section 1 Area



b. Length. LNAV/VNAV FSL is based on 15 seconds at KTAS + 10 Kts tailwind without ATT. Calculate FSL using formula 3-6-5. Section 1 extension length is the same as non-vertically guided procedures [see formula 3-6-2].

Formula 3-6-5. LNAV/VNAV Flat Surface Length

$$\text{FSL} = \frac{(V_{\text{KTAS}} + 10) \times 15}{3600}$$

Where:
V_{KTAS} = calculated KTAS using DA and final approach KIAS

Example:

$$\text{FSL} = \frac{(173.02 + 10) \times 15}{3600}$$
$$\text{FSL} \approx 0.76 \text{ NM}$$

c. Width. The 15-degree OEA splay begins at the outer edge of the final segment secondary area at the base DA point (line C-D). Calculate the distance from course centerline to the primary and secondary boundary of the OEA at any distance from line C-D using formula 3-6-3.

d. Obstacle clearance. The HMAS is determined by subtracting the applicable flat surface ROC for the CAT [see table 3-3-2] and applicable adjustments from the base DA. Where obstacles penetrate the flat surface, raise the DA by the amount of penetration. Where obstacles penetrate the sloping OCS, either calculate the DA adjustment with formula 3-6-6 or apply a higher than standard climb gradient to clear the obstruction [see paragraph 1-3-1.g(2)]. In any case, obstacle mitigation requires the missed approach to be re-evaluated.

Formula 3-6-6. LNAV/VNAV DA Adjustment Value

$$\Delta DA = r \times e^{\left(\frac{p \times MA_{slope} \times \tan(\theta)}{r \times [1 + MA_{slope} \times \tan(\theta)]}\right)} - r$$

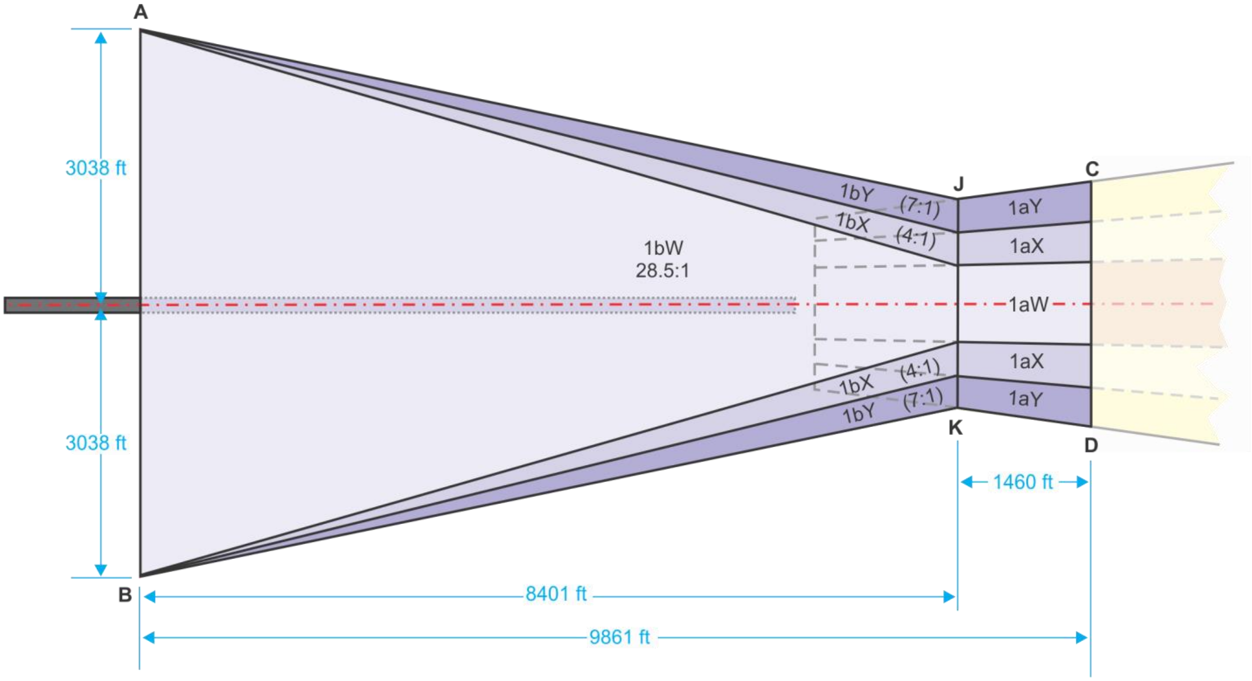
Where:
p = penetration (feet) of the MA OCS
MA_{slope} = missed approach OCS slope ratio

Example:

$$\Delta DA = r \times\ e^{\left(\frac{19.3\ \times 40\ \times\ \tan(3)}{r \times [\,1 + 40\ \times\ \tan\ (3)]}\right)} - r$$
$$\Delta DA \approx 13.07 \text{ feet}$$

3-6-3. LPV/GLS. Section 1 begins at the base DA (line C-D) and ends at the SOC (line A-B). It accommodates height loss and establishment of missed approach climb gradient. Section 1 is centered on a continuation of the final approach track and is subdivided into sections 1a and 1b [see figure 3-6-5].

Figure 3-6-5. LPV/GLS Missed Approach Section 1 Area



a. Section 1a area/OCS elevation. Section 1a is a 1460 feet continuation of the FAS OCS beginning at the base DA point to accommodate height loss. The portion consisting of the continuation of the W surface is identified as section 1aW. The portions consisting of the continuation of the X or Y surfaces are respectively identified as section 1aX or 1aY. Calculate the width and elevation of the section 1aW, 1aX, and 1aY surfaces at any distance from LTP using the final segment formulas.

b. Section 1b areas. The section 1b area extends from line J-K at the end of section 1a for a distance of 8401 feet to line A-B. Section 1b is subdivided into sections 1bW, 1bX, and 1bY. Calculate the distance from course centerline to the boundary of these surfaces at any distance from the end of section 1a using formula 3-6-7.

(1) Section 1bW. Section 1bW extends from the end of section 1aW for a distance of 8401 feet. Its lateral boundaries splay from the width of the end of 1aW surface to a width of ±3038 feet either side of the missed approach course at the 8401 foot-point.

(2) Section 1bX. Section 1bX extends from the end of section 1aX for a distance of 8401 feet. Its inner boundary is the outer boundary of the 1bW surface. Its outer boundary splays from the end of the 1aX surface to a width of ±3038 feet either side of the missed approach course at the 8401-foot point.

(3) Section 1bY. Section 1bY extends from the end of section 1aY for a distance of 8401 feet. Its inner boundary is the outer boundary of the 1bX surface. Its outer boundary splays from the outer edge of the 1aY at the surface at the end of section 1a to a width of ± 3038 feet either side of the missed approach course at the 8401-foot point.

Formula 3-6-7. Section 1bW/X/Y Boundary

$$1\mathrm{b}_{\mathrm{boundary}} = \frac{\mathrm{d}_{1\mathrm{aEnd}} \times (3038 - 1\mathrm{a}_{\mathrm{boundary}})}{8401} + 1\mathrm{a}_{\mathrm{boundary}}$$

Where:
$\mathrm{d}_{1\mathrm{aEnd}}$ = along track distance (feet) from end of section 1a to the point of interest
$1\mathrm{a}_{\mathrm{boundary}}$ = perpendicular dist. (feet) from centerline to respective ending 1a outer boundary

Example:

$$1\mathrm{b}_{\mathrm{boundary}} = \frac{2591.8 \times (3038 - 481.06)}{8401} + 481.06$$
$$1\mathrm{b}_{\mathrm{boundary}} \approx 1269.90 \text{ feet}$$

c. Section 1b OCS elevations.

(1) Section 1bW. The surface rises from the elevation of the 1aW surface at the end of section 1a at a slope ratio of 28.5:1. Calculate the section 1bW surface elevation with formula 1-3-6 for input into formula 3-6-8.

(2) Section 1bX. The surface rises at a slope ration of 4:1 perpendicular to the missed approach course from the edge of the 1bW surface. Calculate the adjustment (Q) for X surface obstacles using formula 3-4-8.

(3) Section 1bY. The surface rises at a slope ratio of 7:1 perpendicular to the missed approach course from the edge of the 1bX surface. Calculate the adjustment (Q) for Y surface obstacles using formula 3-4-10.

Formula 3-6-8. Section 1b Surface Penetration

$$\mathrm{p} = \mathrm{O_{MSL}} - \mathrm{Q} - \mathrm{1bW_{elev}}$$

Where:
Q = obstacle adjustment (feet) for X or Y surface rise. Zero (0) if in the W surface
$\mathrm{1bW_{elev}}$ = elevation (feet) of the 1bW surface abeam the obstacle

Example:

$\mathrm{p} = 1325.8 - 24.22 - 1282.70$
$\mathrm{p} \approx 18.88$ feet

d. Section 1b OCS evaluation. To eliminate or avoid a penetration, take one or more of the following actions:

(1) Removing or adjusting the obstruction location and/or height.

(2) Raising GPA within categorical limits.

(3) Raising TCH [see paragraph 3-4-7].

(4) Adjusting DA. For a surface 1b penetration of p feet, the DA point must move Δd_{DA} [see formula 3-6-9] feet further from the LTP to raise the surface above the penetration.

Formula 3-6-9. Along-track DA Adjustment

$$\Delta \mathrm{d_{DA}} = \frac{\mathrm{p} \times 28.5 \times \mathrm{FAS_{slope}}}{28.5 + \mathrm{FAS_{slope}}}$$

Where:
p = penetration (feet) of the 1b OCS
$\mathrm{FAS_{slope}}$ = final approach segment OCS slope ratio

Example:

$$\Delta \mathrm{d_{DA}} = \frac{18.88 \times 28.5 \times 34}{28.5 + 34}$$
$\Delta \mathrm{d_{DA}} \approx 292.72$ feet

Section 3-7. Missed Approach Section 2

3-7-1. General.

a. Obstacle evaluation area (OEA). The section 2 OEA begins at the end of section 1, and splays at least 15 degrees relative to the nominal track to reach full width. Refer to individual chapters for MA section 1 information.

b. Aircraft start of climb (SOC). When calculating the SOC, use the base MDA/DA value from section 1 [see paragraph 3-5-1].

(1) Non-LPV/GLS. When line J-K and line A-B are not coincident, calculate the SOC by inputting base MDA/DA, CG, and distance between J-K and A-B into formula 1-3-8.

(2) LPV/GLS. Use formula 3-7-1 to calculate the section 1 end altitude (SOC).

Formula 3-7-1. LPV/GLS SOC

$$\mathrm{SOC} = \mathrm{DA} - \tan(\theta) \times 1460 + \frac{8401 \times 0.3048 \times 200}{1852}$$

Example:

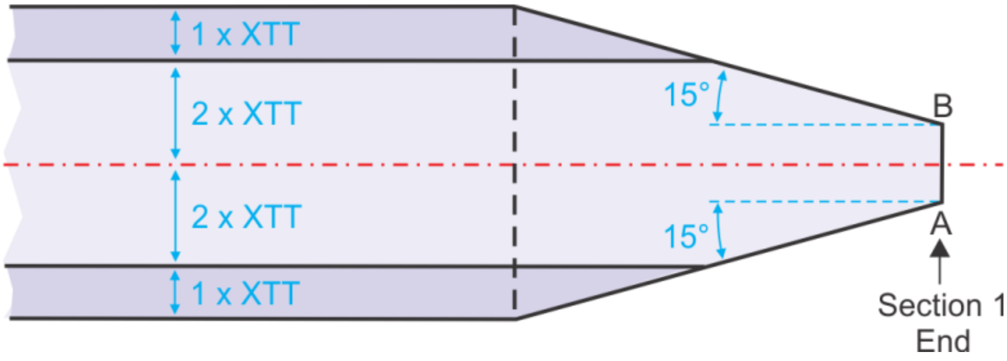
$$\mathrm{SOC} = 1225 - \tan(3.1) \times 1460 + \frac{8401 \times 0.3048 \times 200}{1852}$$

$$\mathrm{SOC} \approx 1422.45 \text{ feet}$$

c. Obstacle clearance surface (OCS) slope and origin. The section 2 OCS slope [see formula 1-3-5] begins at line A-B. Where a higher than standard CG is required, apply the associated OCS from the SOC to the CG termination altitude, then revert to the default OCS.

3-7-2. Straight missed approach. The straight course is a continuation of the FAC. The straight MA section 2 OEA begins at the end of section 1 and splays at 15 degrees relative to the nominal track until reaching full primary and secondary width.

Figure 3-7-1. Missed Approach Section 2 Straight Area



3-7-3. Turning missed approach.

a. Early and inside turn construction.

(1) First turn. The first MA turn may be specified at an altitude or be specified at a FB or FO fix.

(a) Turn-at-altitude. Order 8260.19, chapter 8 paragraph 'Missed approach instructions' applies except a turn altitude must be specified for any amount of turn. The minimum turn altitude must equal or exceed the higher of the end of the section 1 end aircraft altitude (SOC) or 400 feet above airport elevation. A turn-at-altitude is followed by a DF leg usually ending with a DF-TF connection or holding pattern entry.

1. Turn initiation area (TIA). The TIA includes the applicable section 1 area starting at line C-D and ends where the specified turn altitude is reached (line L-L'). Construct the TIA from the end of section 1 to line L-L' using straight MA criteria. Evaluate section 2 obstacles by applying the prescribed OCS slope along the shortest distance from the TIA boundary to the obstacle. The obstacle-based turn altitude is the sum of starting ROC and OCS elevation at the TIA boundary needed to clear section 2 obstacles. Because TIA length and the OCS elevation at the end of the TIA is dependent on aircraft altitude, establishing an obstacle-based turn altitude may be an iterative process. Calculate lengths from line C-D using formula 3-7-2, formula 3-7-3, formula 3-7-4, and formula 3-7-5.

a Non-LPV/GLS TIA. Calculate the TIA length using the appropriate FSL value [see section 3-5.]. Where an increased CG terminates below the turn altitude, apply formula 3-7-2, otherwise apply formula 3-7-3.

Formula 3-7-2. TIA Length CG Termination Below Turn Altitude (Non LPV/GLS)

$$\mathrm{TIA_{length}} = \frac{\mathrm{FSL} \times \mathrm{r}}{\mathrm{r} + \mathrm{alt_b}} + \mathrm{D_{climb_i}} + \mathrm{D_{climb_{ii}}}$$

Where:
FSL = length (NM) of the missed approach flat surface
$\mathrm{alt_b}$ = beginning altitude (feet) (i.e., the MDA for LNAV/LP or DA for LNAV/VNAV)
$\mathrm{D_{climb_i}}$ = climb distance (NM) required to reach the first $\mathrm{CG_{term}}$ from MDA or DA
$\mathrm{D_{climb_{ii}}}$ = climb distance (NM) required to reach the turn altitude from the first $\mathrm{CG_{term}}$

Example:

$$\mathrm{TIA_{length}} = \frac{1.21 \times \mathrm{r}}{\mathrm{r} + 1900} + 4.44 + 3.26$$
$$\mathrm{TIA_{length}} \approx 8.91\ \mathrm{NM}$$

Formula 3-7-3. TIA Length Standard CG or CG Termination at or above Turn Altitude (Non-LPV/GLS)

$$\mathrm{TIA_{length}} = \frac{\mathrm{FSL} \times \mathrm{r}}{\mathrm{r} + \mathrm{alt_b}} + \mathrm{D_{climb}}$$

Where:
FSL = length (NM)of the missed approach flat surface
$\mathrm{alt_b}$ = beginning altitude (feet) (i.e., either the MDA for LNAV/LP or DA for LNAV/VNAV)
$\mathrm{D_{climb}}$ = climb distance (NM) required to reach the turn altitude from MDA or DA

Example:

$$\mathrm{TIA_{length}} = \frac{1.21 \times \mathrm{r}}{\mathrm{r} + 1900} + 6.67$$
$$\mathrm{TIA_{length}} \approx 7.88\ \mathrm{NM}$$

b LPV/GLS TIA. Where an increased CG terminates below the turn altitude, apply formula 3-7-4, otherwise apply formula 3-7-5.

Formula 3-7-4. TIA Length CG Termination Below Turn Altitude (LPV/GLS)

$$\mathrm{TIA_{length}} = \frac{9861 \times 0.3048}{1852} + \mathrm{D_{climb_i}} + \mathrm{D_{climb_{ii}}}$$

Where:
$\mathrm{D_{climb_i}}$ = climb distance (NM) required to reach the first $\mathrm{CG_{term}}$ from SOC
$\mathrm{D_{climb_{ii}}}$ = climb distance (NM) required to reach the turn altitude from the first $\mathrm{CG_{term}}$

Example:

$$\mathrm{TIA_{length}} = \frac{9861 \times 0.3048}{1852} + 3.79 + 3.26$$
$$\mathrm{TIA_{length}} \approx 8.67\ \mathrm{NM}$$

Formula 3-7-5. TIA Length Standard CG or CG Termination at or above Turn Altitude (LPV/GLS)

$$\mathrm{TIA_{length}} = \frac{9861 \times 0.3048}{1852} + \mathrm{D_{climb}}$$

Where:
$\mathrm{D_{climb}}$ = climb distance (NM) required to reach the turn altitude from SOC

Example:

$$\mathrm{TIA_{length}} = \frac{9861 \times 0.3048}{1852} + 6.01$$
$$\mathrm{TIA_{length}} \approx 7.63\ \mathrm{NM}$$

2. OEA construction after TIA [see table 3-7-1 and figure 3-7-2].

Table 3-7-1. Early/Inside Turn Tie-Back Points

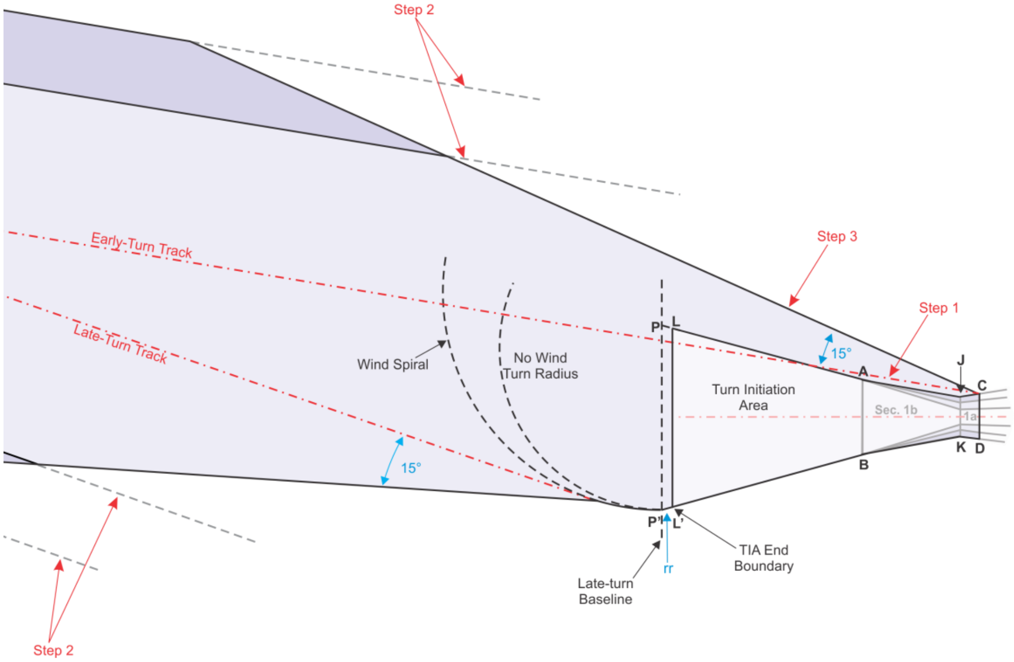
Turn Angle (measured from FAC/line C-D intersection)	Tie-Back Point
≤ 75 degrees	Point C
> 75 degrees	Point D
> 165 degrees	Point P’

a *Step 1*. Construct a line (representing the early-turn flight track) from the tie-back point, to the fix.

b *Step 2***.** Construct the inside turn primary and secondary OEA boundary lines parallel to this line.

c *Step 3*. From the tie-back point, construct a line slaying at 15 degrees to intersect the parallel boundary lines or leg end, whichever occurs earlier. Apply inside turn secondary areas only after the 15-degree splay line intersects the primary boundary line.

Figure 3-7-2. Early/Inside Turn Turn-at-Altitude Construction after TIA



(b) Turn-at-fix (FO or FB). TF legs are followed by a TF leg ending with holding pattern entry or TF-TF connection when the initial straight leg is less than full width. May be followed by an RF leg when the initial straight leg has reached full width, ending with an RF-TF or RF-RF connection. The recommended turn magnitude is 70 degrees; the maximum is 90 degrees. Use FB unless a FO is required for obstacle avoidance or where mandated by specific operational requirements. Use paragraph 1-2-5.d in cases where full primary and secondary width is achieved at a sufficient distance prior to the turn fix to permit construction. Otherwise, construct as follows:

1. Fix location. The first turn fix must be located on the final approach track extended. The distance to the turn fix must be sufficient to result in an early turn baseline at or beyond the end of section 1.Where the first fix must be located at the point the aircraft reaches or exceeds a specific altitude, calculate fix distance from line A-B with formula 1-3-9 using the standard or increased CG. If a fix is already established, calculate the aircraft altitude at the fix using formula 1-3-8. Calculate the aircraft SOC per paragraph 3-7-1.b.

2. Early turn baseline (line L – L'). The early turn baseline is established prior to the turn fix perpendicular to the inbound leg at the appropriate table 3-7-2 distance.

Table 3-7-2. Early turn Baseline Distance from Turn Fix

Fix Type	Distance
FB Fix	ATT + DTA*
FO Fix	ATT

*DTA = 0 for turns 10 degrees or less

3. Construction points. Inside turn construction uses points PI (primary intersection) and SI (secondary intersection). When defining these points, the leg prior to the turn fix is the inbound leg and the leg following the turn fix is the outbound leg.

a Point SI. This is the connection point for the secondary area boundary and is the intersection of line L-L' and the inbound leg secondary area boundary. When inbound leg has no secondary, points PI and SI are coincident.

b Point PI. This is the connection point for the primary area boundary and is the intersection of line L-L' and the inbound leg primary area boundary.

4. Inside turn construction. Primary/Secondary boundary expansion depends on the location of points PI and SI in relation to the outbound leg OEA boundary lines (extended when necessary) [see table 3-7-3 and figure 3-7-3, figure 3-7-4, and figure 3-7-5]. Where standard expansion is suitable for one, but not both splays, find the outbound connection point for the non-standard splay abeam the standard connection point. If the table 3-7-3 expansion line results in smaller primary areas than those resulting from the 15-degree inbound leg splay, relocate point PI and/or increase the splay angle to maintain or increase the size of the inbound and outbound leg primary areas. Construct as follows:

Table 3-7-3. Inside Turn Expansion

Construction Point	External to Outbound leg OEA (both primary and secondary)	Internal to Outbound leg Secondary area	Internal to Outbound leg Primary area
Point SI	Greater of ½ Turn-angle or Inbound leg Splay	15 degrees relative outbound track	15 degrees relative outbound track
Point PI	Greater of ½ Turn-angle or Inbound leg Splay	Greater of ½ Turn-angle or Inbound leg Splay	15 degrees relative outbound track

a *Step 1*. Construct line L-L' at the appropriate table 3-7-2 distance.

b *Step 2*. Construct secondary area boundary.

1 Case 1. SI is external to the outbound leg OEA. Construct an expansion line from SI to the outbound leg secondary boundary.

2 Case 2. SI is internal to either the outbound leg secondary or primary area. In this case, an alternative construction point (point SI') may be required depending on turn magnitude. Point SI' is defined as the intersection of line L-L' and the outside-turn secondary area boundary. Construct the expansion line from either point SI or SI' to the outbound leg secondary boundary whichever results in the larger area. When the expansion line reaches the outbound secondary boundary before reaching the point of intersection of the inbound and outbound leg secondary boundaries, no expansion is required and the area is a simple connection of the inbound and outbound leg secondary boundaries.

c *Step 3*. Construct the primary area boundary. Construct an expansion line from point PI to the outbound leg primary boundary. When the *Step 2* expansion line reaches the outbound leg primary boundary before reaching point of intersection of the inbound and outbound leg primary boundaries, no expansion is required and the area is a simple connection of the inbound and outbound leg primary boundaries.

Figure 3-7-3. Early/Inside Turn, First Turn, Turn at Fix: SI/PI Collocated and External to Outbound Primary and Secondary

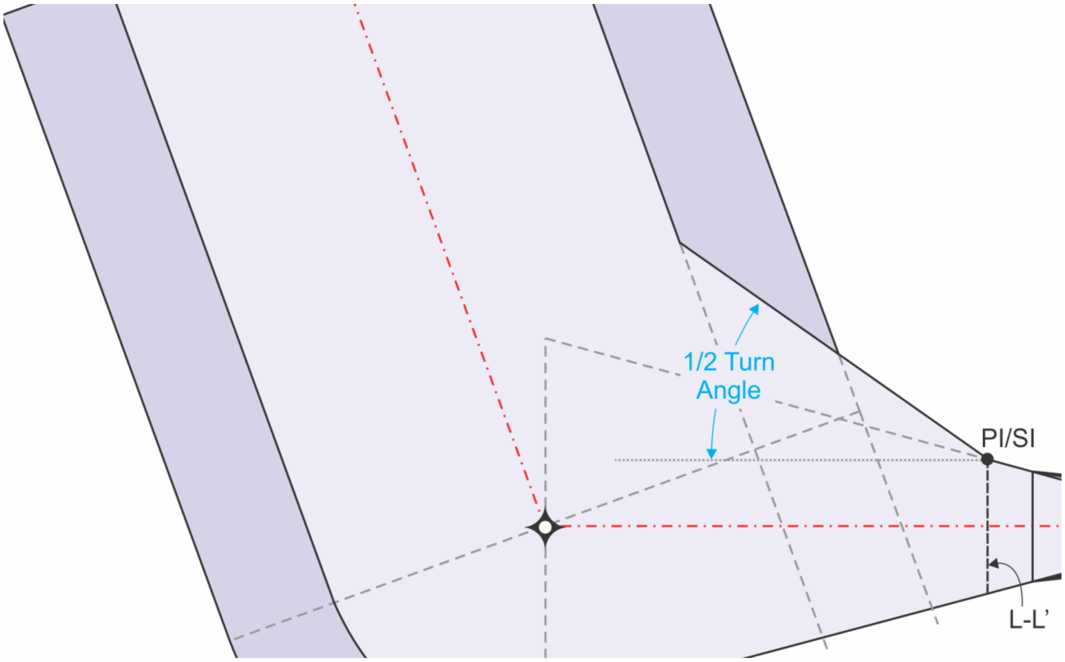


Figure 3-7-4. Early Inside Turn, First Turn, Turn at Fix: SI/PI internal to Outbound Primary

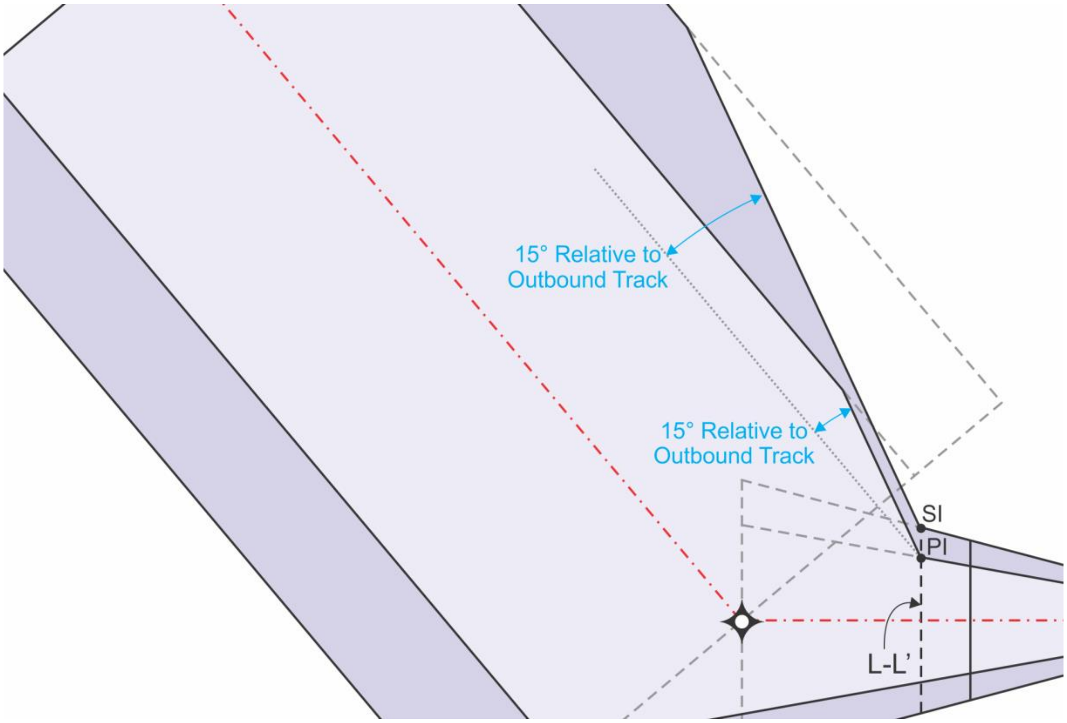
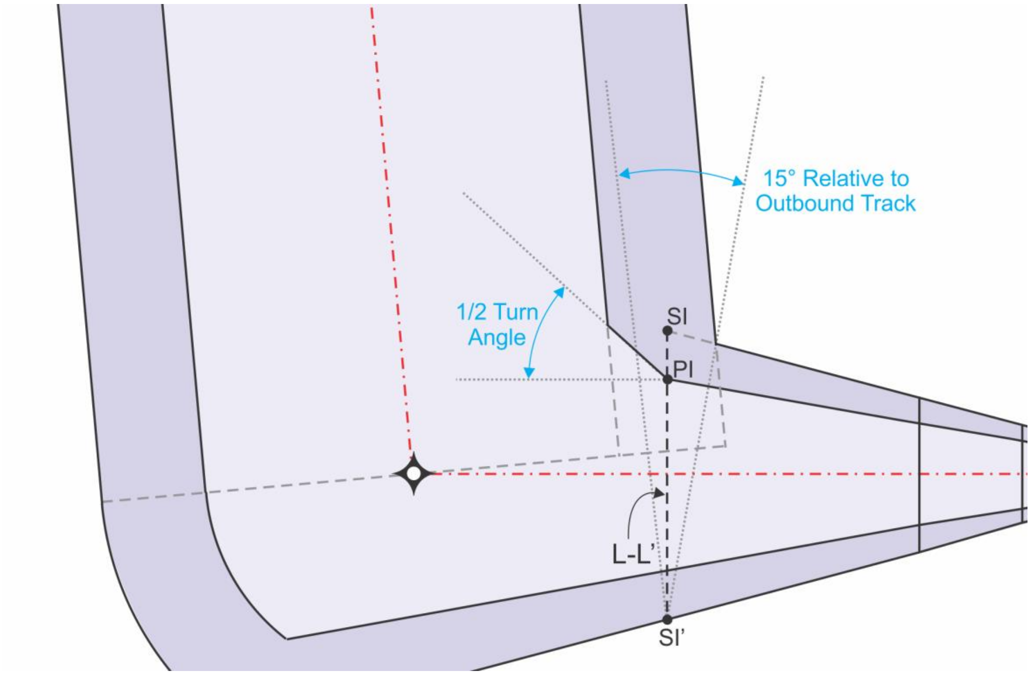


Figure 3-7-5. Early/Inside Turn, First Turn, Turn at Fix:

SI and PI internal to Outbound Secondary, splay from point SI'



(2) Second turn. The second MA turn is specified at either a FB or FO fix [see figure 3-7-6].

(a) DF-TF. This applies to a DF-TF connection following a turn-at-altitude.

1. Construct in accordance with paragraph 3-7-3.a(1)(b) with the following exceptions:

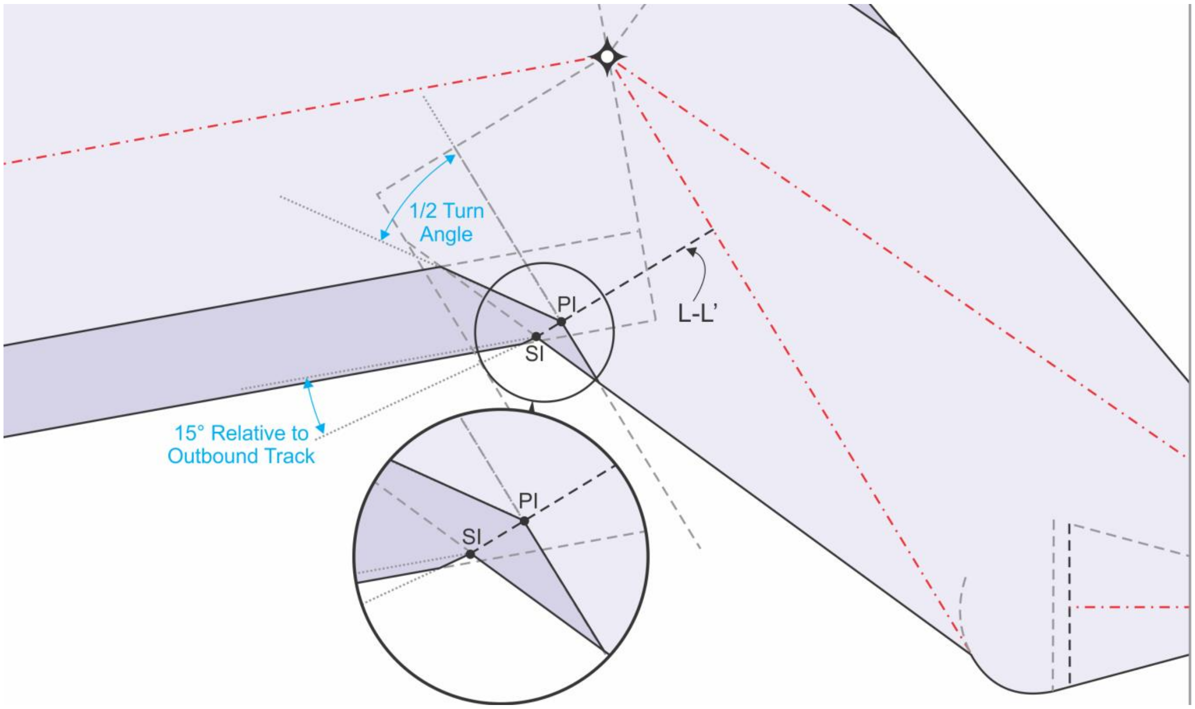
a Paragraph 3-7-3.a(1)(b)1 does not apply.

b Line L-L' is established perpendicular either on the early- or late-turn track, whichever is on the inside of the turn.

c When full primary- and secondary-width is reached at the early-turn baseline (L-L') construct in accordance with paragraph 1-2-5.d.

(b) Other than DF–TF connections. Construct in accordance with paragraph 1-2-5.d.

Figure 3-7-6. Early/Inside Turn, Second Turn, Turn at FB Fix DF-TF connection



b. Late and outside turn construction.

(1) Wind spiral application. Wind spiral (WS) construction applies to late and outside turns for turn-at-altitude, turn-at-fix (FO) for the first turn, and DF-TF (FO) for the second turn. See paragraph 1-3-1.g for design parameters.

(a) WS number and baselines. Baseline locations for WS construction may be dependent on Reaction and Roll distance [see formula 1-2-12 and table 3-7-4].

Table 3-7-4. Wind Spiral Number and Distance to Late Turn Baseline

WS/Baseline Parameters	Turn-at-Fix (FO)	Turn-at-Altitude
WS Number	1 or 2	1, 2, or 3
WS1 & WS2 Distance Late-Turn Baseline from turn fix/point	ATT+D_{rr}	TIA+ D_{rr}
WS3 Baseline	Not applicable	Parallel to line C-D

1. First turn baseline. For first turn construction, the late-turn baseline (line P-P') marks the construction line for wind spirals. Line P-P' is located at the table 3-7-4 distance with points P and P' placed at the continuation of the inbound leg's outer boundary [see paragraph 1-3-1.g(4)]. The no-wind turn radius for the outside turn wind spiral (WS1) and inside turn wind spiral (WS2) will be measured perpendicular to the inbound track from their respective point P. If a third wind spiral is used (WS3), the baseline will be parallel to line C-D with the no-wind turn radius measured inward from point C.

2. Second turn baseline [see figure 3-7-14]. To accommodate both inbound tracks, the second MA turn construction uses two WS baselines, line P'-P" for WS1 and line P-P' for WS2. Each late turn baseline is oriented perpendicular to the early- and late-turn tracks at the table 3-7-4 distance. The baseline for the inbound track nearer the outside-turn boundary is designated line P'-P", with point P" placed on the extended outer boundary and point P' placed at the no-wind turn radius inward along the baseline from point P". The baseline for the inbound track nearer the inside-turn boundary is designated line P-P', with point P' placed on inbound track and point P placed on the extended outer boundary. The no-wind turn radius for the inside turn wind spiral is measured inward along the baseline from point P.

(b) WS connection point and outside turn OEA boundary [see figure 3-7-7]. Each WS has various connection options along its path, which predicate the outside turn OEA boundary. The chosen connection must provide the most reasonably conservative track and protection area. Where excessive splay is required to reach full-width protection, consider lengthening the leg, restricting the speed, category, etc. to avoid protection and/or construction difficulties. Consider full-width protection to exist at the fix where the splay line is tangent to a full-width-radius circle about the fix [see figure 3-7-10].

1. Turn-at-altitude connection point. For turn- at- altitude, the 15-degree or greater splay line that joins the outbound leg outer boundaries may originate from the WS/direct-to-fix tangent point (point 1), the WS to WS tangent line origin (point 2) or the WS to WS tangent line end (point 3). Where the turn angle is ≤ 105 degrees, or the divergence angle between the WS to WS tangent line and the direct-to-fix line is ≤ 15 degrees, apply the splay line from the WS to WS tangent line origin. DF secondary areas begin/exist only where full width primary exists.

2. Turn-at-fix (FO) Connection Point. For turns at a fix, the connection point and the OEA boundary is dependent on the WS boundary relative to the outbound leg OEA (extended). The connection point is the point of tangency on the WS in both cases. When the WS is not contained in the outbound leg OEA, the outside boundary is a 30-degree converging line relative to the outbound course [see figure 3-7-8].When the WS is contained in the outbound leg OEA, the outside boundary is a 15-degree or greater splaying line relative to the outbound course from the WS/outbound leg parallel point until reaching the outbound leg boundaries [see figure 3-7-9].

Figure 3-7-7. Late/Outside Turn, WS Connection Points and Outside Turn OEA Boundary

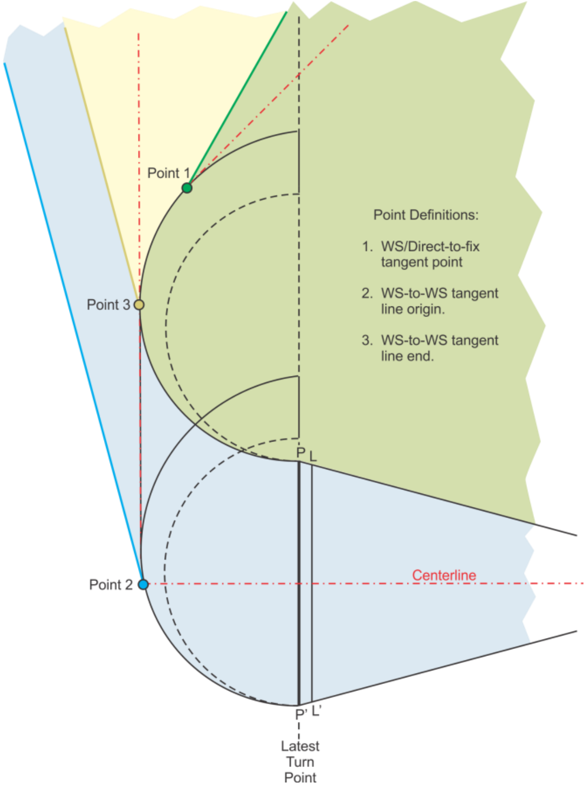

Point 1
Point 3
Point 2
Point Definitions:
1. WS/Direct-to-fix tangent point
2. WS-to-WS tangent line origin.
3. WS-to-WS tangent line end.
P
L
Centerline
P'
L'
Latest Turn Point

Figure 3-7-8. Late/Outside Turn, Wind Spiral, WS not contained in Outbound OEA

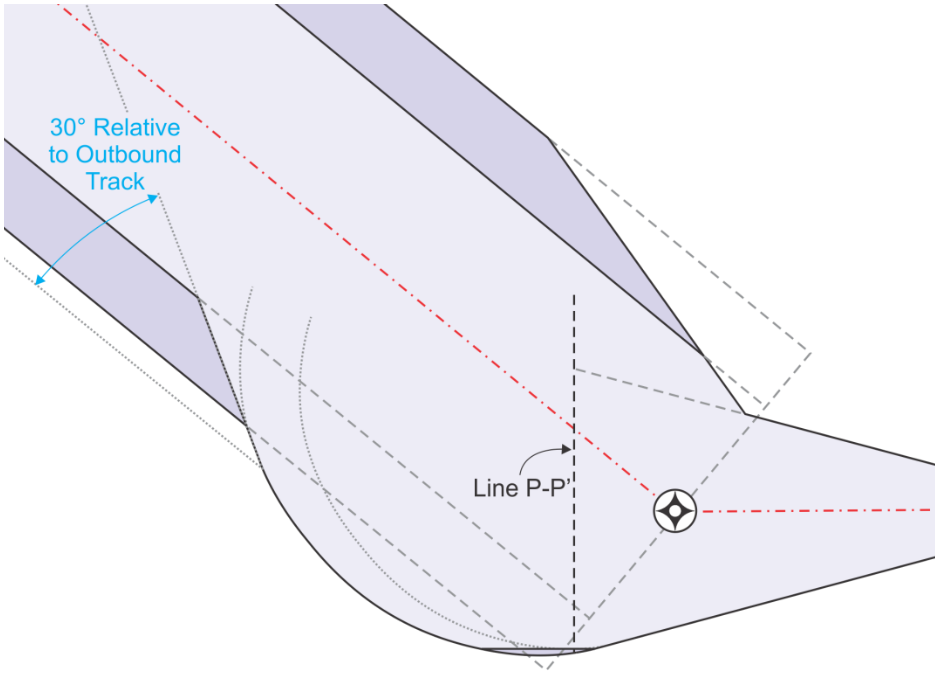
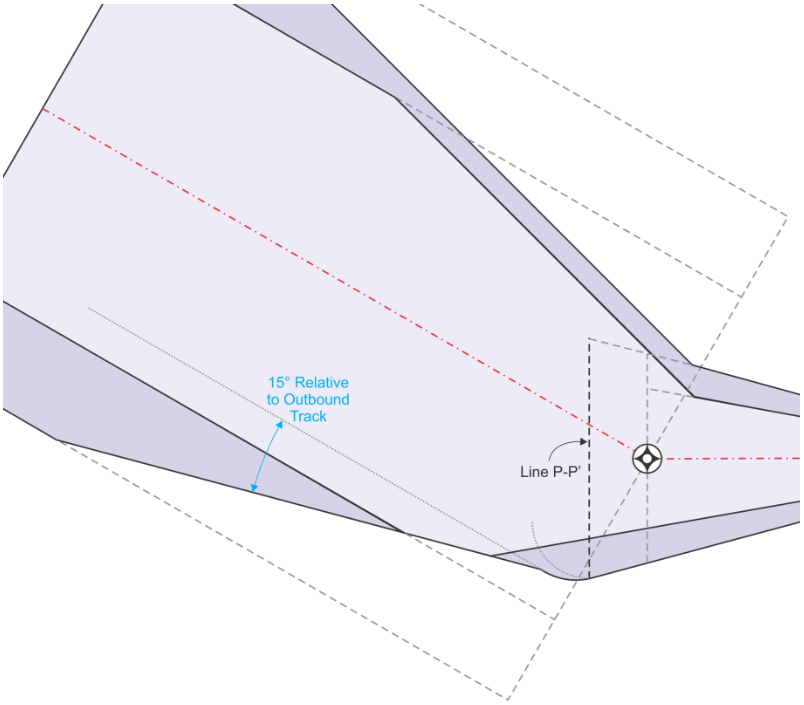


Figure 3-7-9. Late/Outside Turn, Wind Spiral, WS contained in Outbound OEA



(c) Determining multiple WS necessity. To determine multiple wind spiral necessity, construct the additional WS in the direction of turn from its prescribed location to its connection point. Where the additional WS intersects the preceding WS construction (including the connecting and expansion lines), connect the wind spirals with a tangent line that is parallel to the WS center points. Otherwise, revert to the previous WS construction.

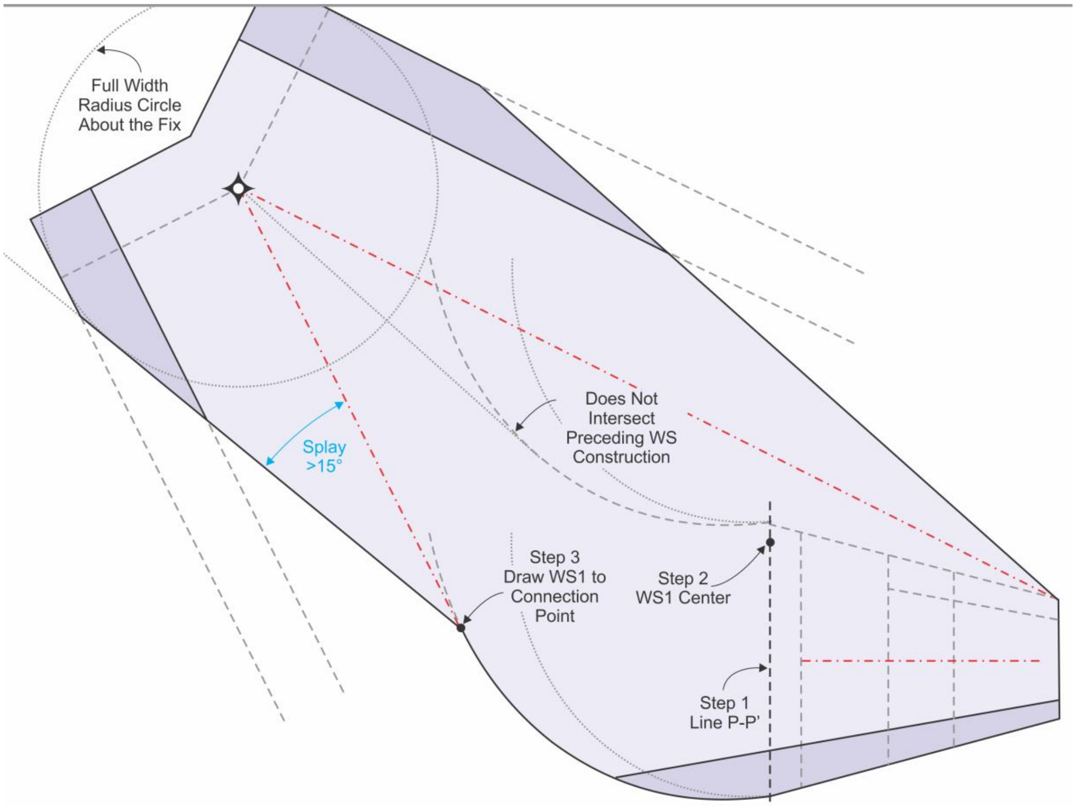
(2) First turn.

(a) Turn-at-altitude or turn-at-fix (FO).

1. *Step 1*. Construct the late-turn baseline (P-P') perpendicular to the straight missed approach track at the late-turn-point.

2. *Case 1*. Small turns using one WS [see figure 3-7-10].

Figure 3-7-10. Late/Outside Turn, First Turn, Case 1,Turn-at-Altitude or FO fix with 1 WS



a *Step 2*. Locate the WS1 center on line P-P' at no-wind turn radius distance from point P'.

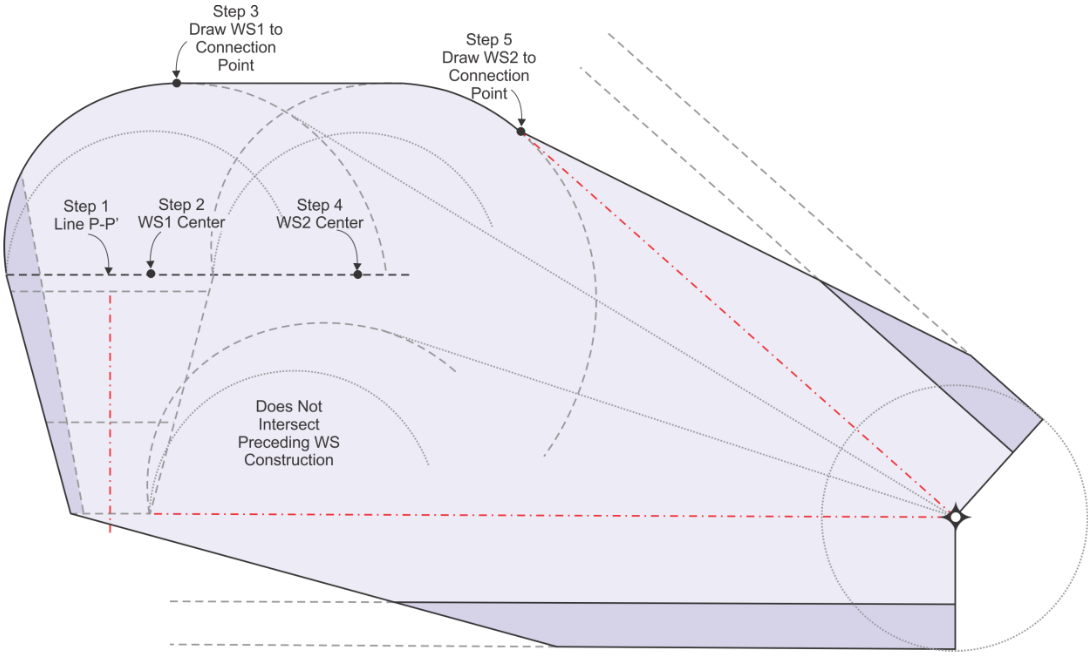
b *Step 3*. Construct WS1 from the outside-turn point in the direction of turn until reaching its connection point.

3. *Case 2*. Turns nearing or greater than 90 degrees using more than one WS [figure 3-7-11]. Apply *Steps 1-3* above, then:

a *Step 4*. Locate the WS2 center on P-P' at no-wind turn radius distance from point P.

b *Step 5*. Construct WS2 from the inside turn point in the direction of turn until reaching its connection point.

Figure 3-7-11. Late/Outside Turn, First Turn, Case 2, Turn-at-Altitude or FO fix with 2 WS



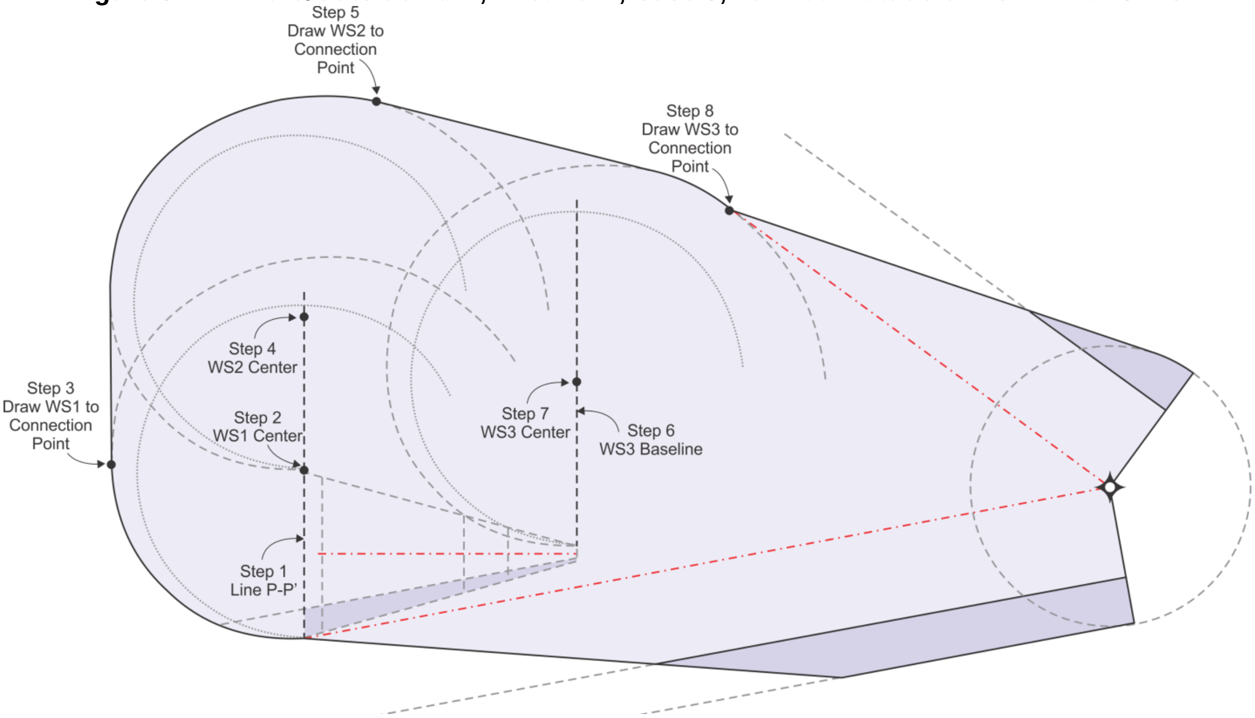
4. *Case 3*. Turns nearing or greater than 180 degrees using more than two WS [see figure 3-7-12]. Apply *Steps 1-5* above, then:

a *Step 6*. Construct the WS3 baseline perpendicular to the straight missed approach track along line C-D extended toward the turn side.

b *Step 7*.Locate the WS3 center on the baseline at no-wind turn radius distance from point C.

c *Step 8*. Construct WS3 from this point in the direction of turn until reaching its connection point.

Figure 3-7-12. Late/Outside Turn, First Turn, Case 3, Turn-at-Altitude or FO fix with 3 WS

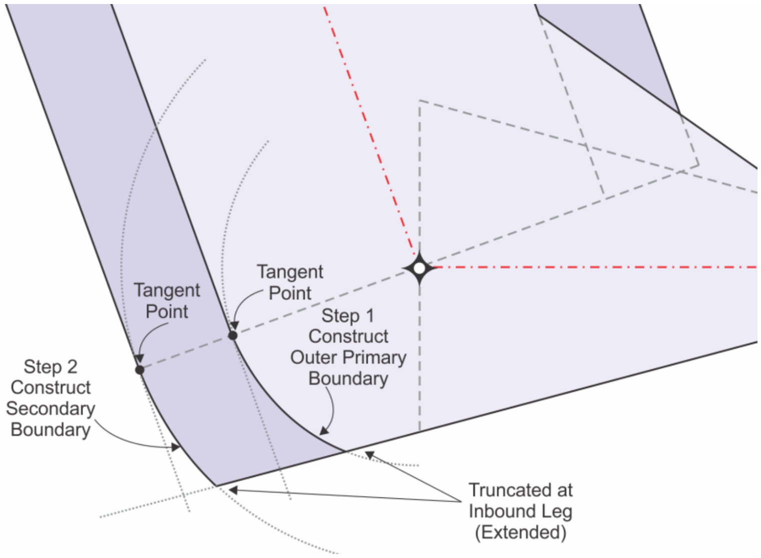


(b) Turn-at-fix (FB) [see figure 3-7-13].

1. *Step 1*. Construct the outer primary boundary using a radius equal to 2 × segment XTT centered on the plotted fix position, truncated at the inbound leg extended outer boundary until tangent to the outbound leg primary boundary.

2. *Step 2*. Construct the secondary boundary using a radius equal to 3 × segment XTT centered on the plotted fix position, truncated at the inbound leg extended outer boundary until tangent to the outbound leg secondary boundary.

Figure 3-7-13. Late/Outside Turn, First Turn, Turn at Fix (FB)


Tangent Point
Tangent Point
Step 1 Construct Outer Primary Boundary
Step 2 Construct Secondary Boundary
Truncated at Inbound Leg (Extended)

(3) Second turn.

(a) Turn-at-fix (FO) [see figure 3-7-14].

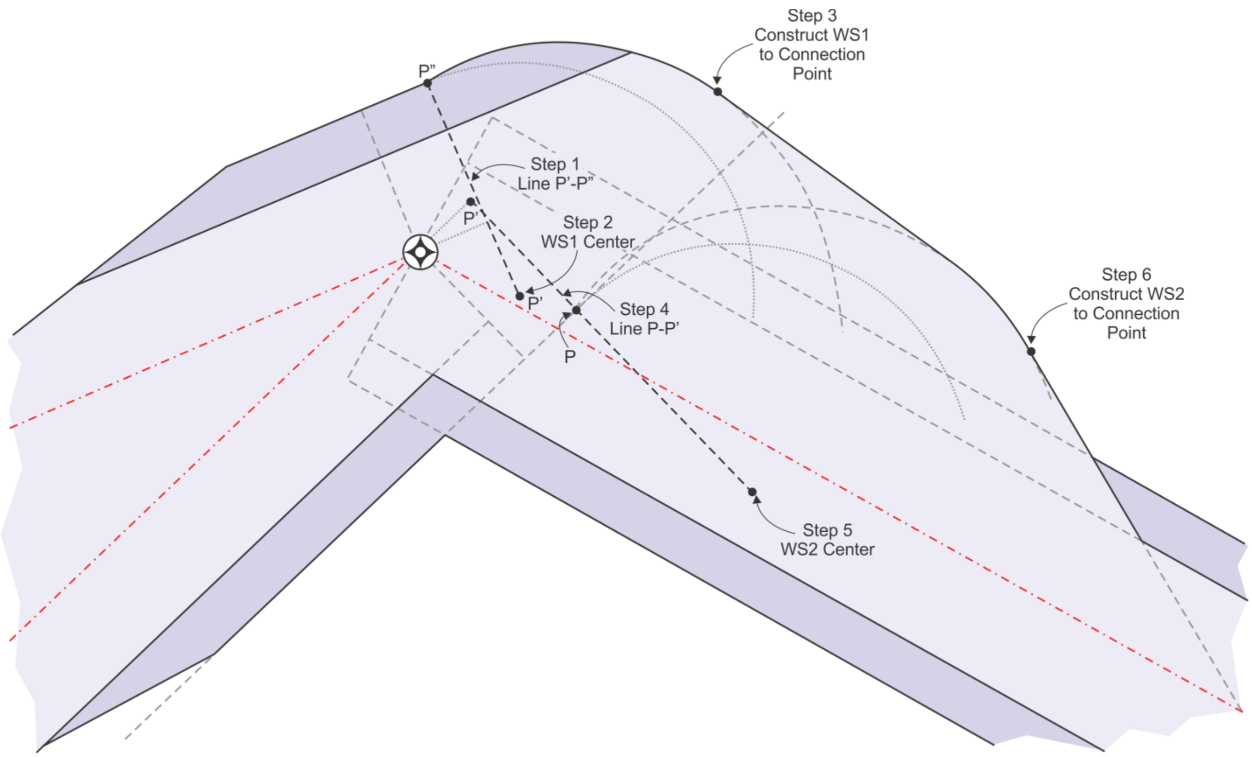
1. *Step 1*. Construct the WS1 baseline, (line P'-P") perpendicular to the DF track nearer the outside of the DF-TF turn, at the late turn point.

2. *Step 2*. Locate the WS1 center on line P'-P" at no-wind turn radius distance from point P."

3. *Step 3*. Construct WS1 from the outside turn point in the direction of turn until reaching its connection point.

4. *Step 4*. Construct the WS2 baseline, (line P-P') perpendicular to the DF track nearer the inside of the DF-TF turn, at the late turn point.

Figure 3-7-14. Late/Outside Turn, Second Turn, Turn-at fix, FO



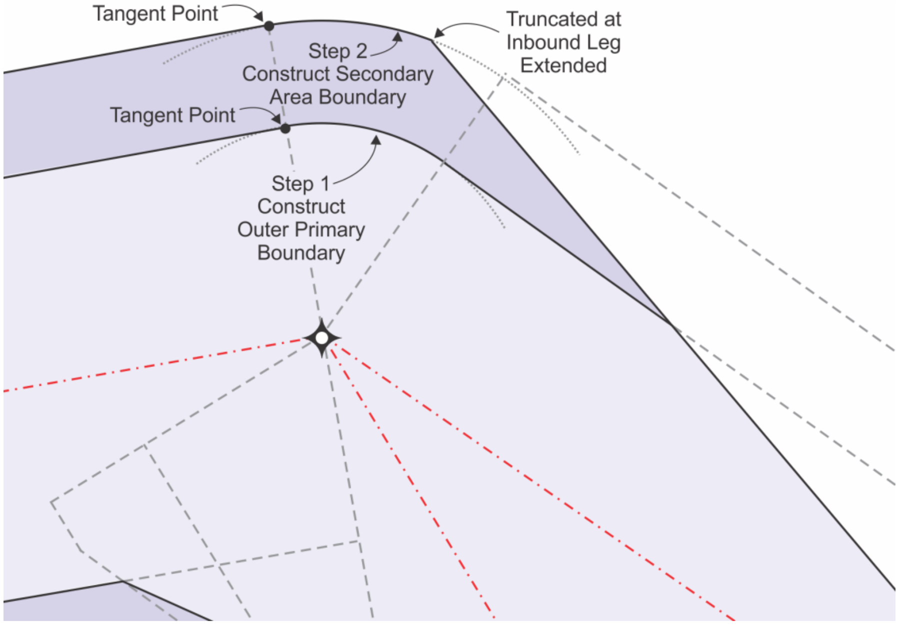
5. *Step 5*. Locate the WS2 center on line P-P’ at no-wind turn radius distance from point P.

6. *Step 6*. Construct WS2 from the inside turn point in the direction of turn until reaching its connection point.

(b) Turn-at-fix (FB).

1. When the inbound leg outside boundary is less than full primary and secondary width reached perpendicular to the fix, construct in accordance with paragraph 3-7-3.b(2)(b) [see figure 3-7-15].

Figure 3-7-15. Late/Outside, Second Turn, Turn at FB fix, less than full OEA width



2. When the inbound leg outside boundary is full primary and secondary width perpendicular to the fix, construct in accordance with paragraph 1-2-5.d.

3-7-4. Obstacle Clearance Surface. Apply a sloping OCS [see paragraph 1-3-1.g and paragraph 3-7-1.c]. Calculate the aircraft SOC in accordance with paragraph 3-7-1.b. Where multiple measurements are required (e.g., a point of interest is equidistant from multiple primary boundaries, it lies along perpendiculars from multiple primary boundaries, etc.), apply the most adverse result from each of the combined primary/secondary measurements.

a. Primary area.

(1) Straight MA/TIA. Measure and apply the OCS slope from line A-B along the track distance to the point or abeam the point of interest [see figure 3-7-16].

(2) Turn-at-altitude (single and multiple legs). Apply the OCS slope along the shortest primary area distance from the TIA boundary to the point of interest [see figure 3-7-16 and figure 3-7-17, and figure 3-7-18].

(3) Turn-at-a-fix (single and multiple legs). Apply the OCS slope from line A-B (parallel to track) to L-L’, then along the shortest primary distance to the point of interest [see figure 3-7-19].

Note: The shortest primary area distance is the length of the shortest line kept within the primary area that passes through the early-turn baseline of all preceding legs.

Figure 3-7-16. OCS, Primary Area, Turn -at-Altitude, Single leg, Small Turn

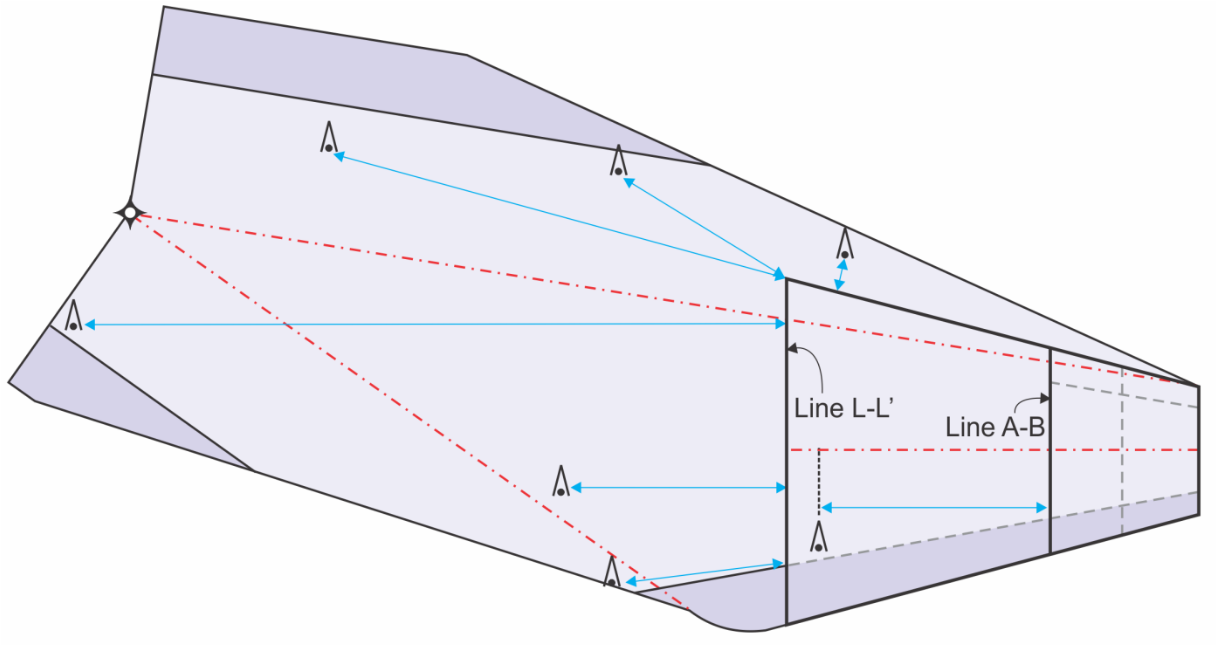


Figure 3-7-17. OCS Primary Area, Turn-at-altitude, Single Leg, Large Turn

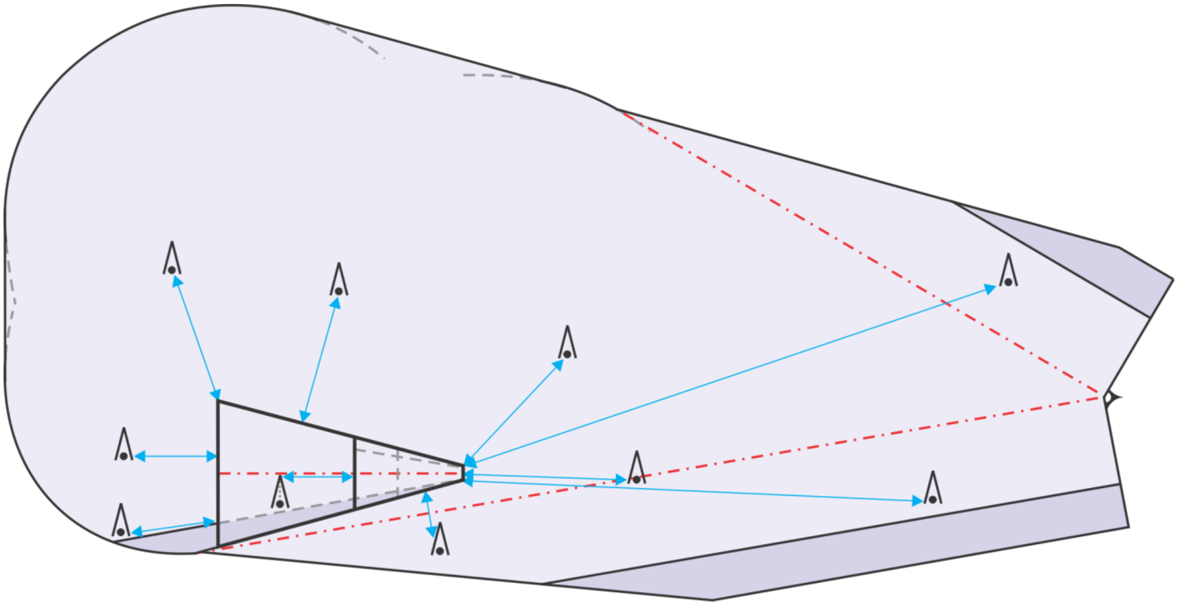


Figure 3-7-18. OCS, Primary Area, Turn-at-Altitude, Multiple legs

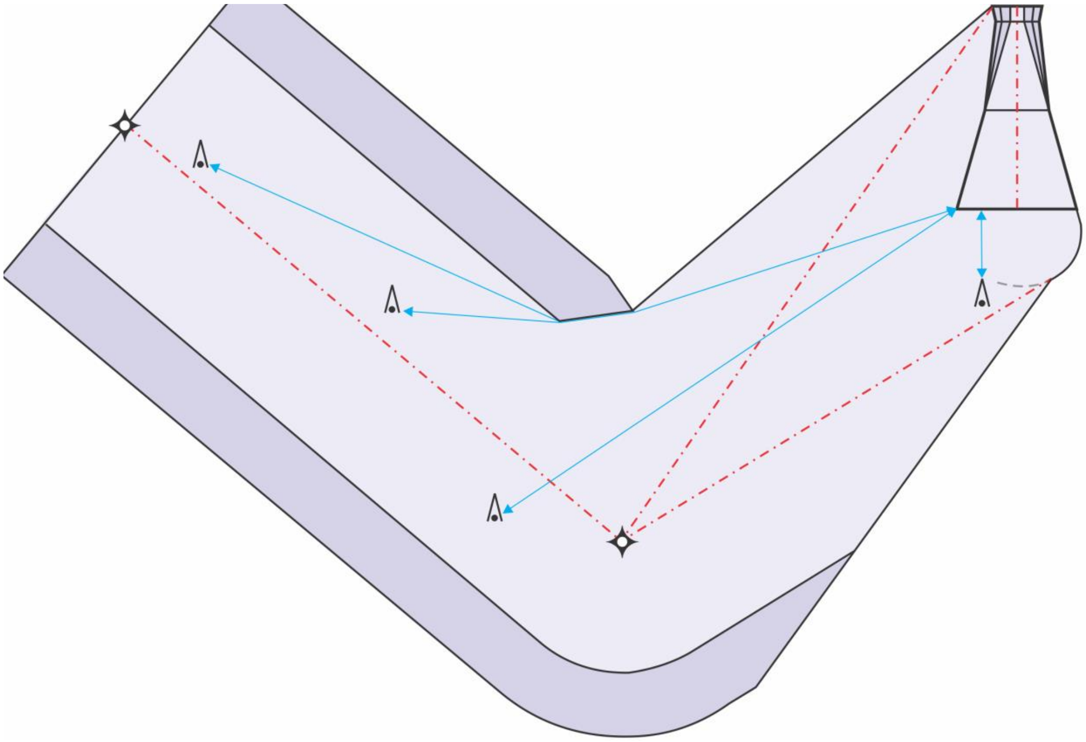
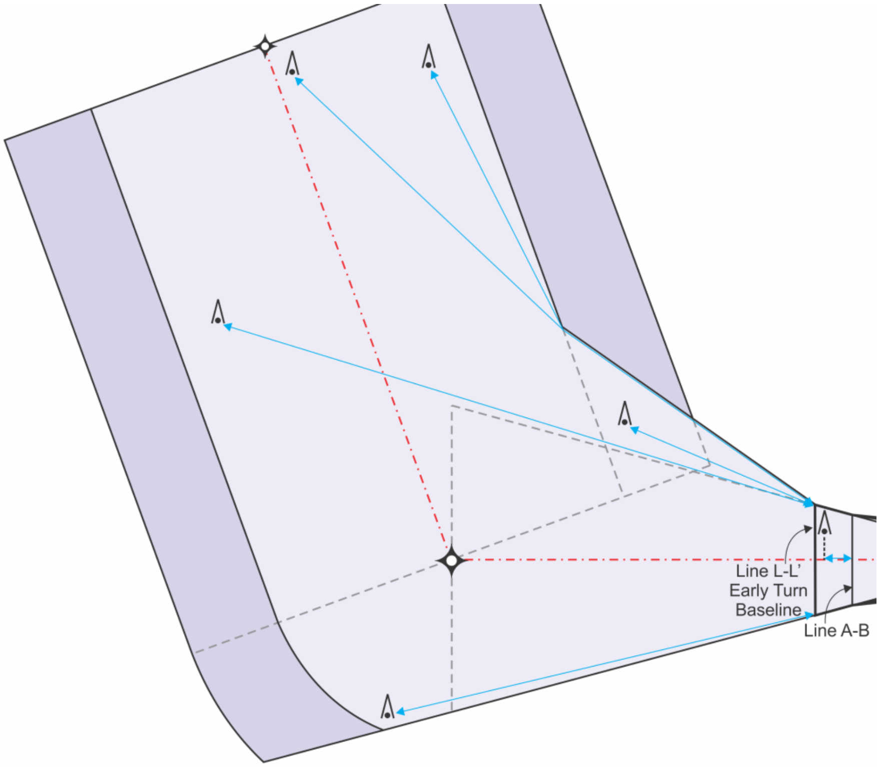
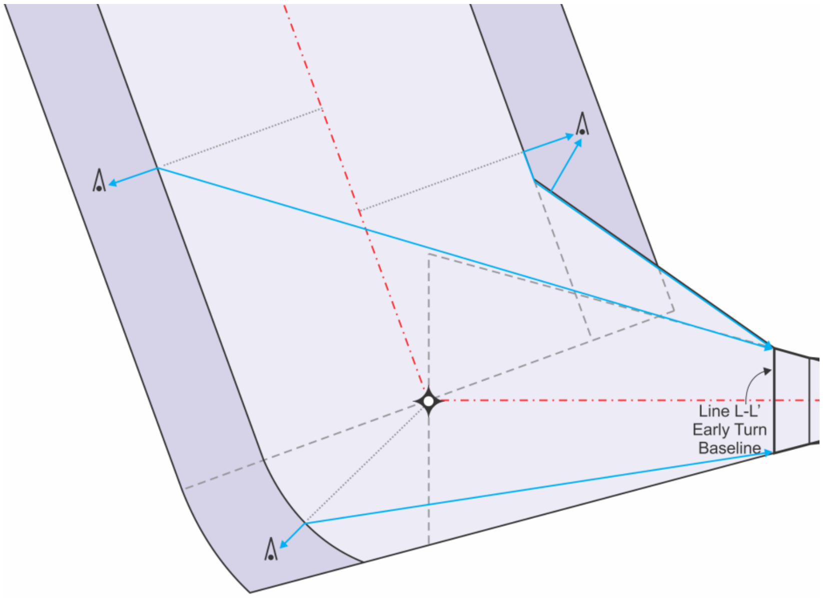


Figure 3-7-19. OCS, Primary Area, Turn at Fix



b. Secondary area. For secondary areas, calculate the primary OCS elevation as stated then apply a 12:1 OCS slope along the shortest secondary distance to the point of interest. In straight legs, this is normally perpendicular to the nominal track. In expansion areas, the slope rises in a direction perpendicular to the primary boundary (arc, diagonal corner-cutter, etc.) [see figure 3-7-20].

Figure 3-7-20. OCS, Secondary Area



THIS PAGE IS INTENTIONALLY LEFT BLANK

Chapter 4. Standard for Required Navigation Performance (RNP) Approach Procedures with Authorization Required (AR)

Section 4-1. General Criteria

4-1-1. Concept and design. TF legs are preferred but RF legs may be used for turn path control, procedure simplification, or improved flyability. FO fixes that require turn construction are not authorized in RNP AR APCH segments.

a. OEA construction. Apply paragraph 1-2-5 with the following exceptions.

(1) Apply the largest RNP for the flight phase from table 1-2-1 unless a smaller value is required to achieve a desired ground track or is operationally required.

(2) RNP changes must occur at a fix and are linear (i.e., do not splay/taper 30 degrees relative to the course). RNP may be increased or decreased as needed prior to the FAS, e.g., where use of a bank angle greater than 20 degrees is needed. RNP changes must not occur in the FAS. After crossing the LTP/FTP, RNP values may only increase. See figure 1-2-1, figure 1-2-2, and figure 1-2-3 for illustrations of RNP changes.

(3) The length of any leg between the initial approach fix and the missed approach point may be reduced to not less than 0.2 NM (regardless of RNP) when there are no more than three waypoints within 1 NM along-track distance and there is no leg shorter than 1 NM prior to a change in RF turn direction. Avoid establishing reduced length legs in sequence to avoid negative impact on chart/display readability.

b. Use of NAVAIDS. Do not incorporate a VOR/DME or VORTAC into an RNP AR procedure if the geodetic coordinates of the VOR and DME source are not identical to 0.01 seconds, regardless whether the facilities are considered collocated.

c. Lines of minima. No more than four lines may be established. Circling minimums are not established.

(1) Establish a default line. The default line is based on RNP 0.3 in the FAS and a default RNP AR APCH [see paragraph 4-3-1.a(1)] or RNAV 1 [see paragraph 4-3-1.a(2)] missed approach.

(2) When the HAT value of the default line is ≥ 300 or no-lights visibility ≥ 1 SM.

(a) Additional lines may be established when at least a 50-foot reduction in HAT or ¼ SM reduction in visibility can be achieved using one or more of the following;

1. Reduced FAS RNP (values < 0.3 but ≥ 0.10).

2. Reduced RNP MAS [see paragraph 4-3-1.a(3)].

3. MAS CG.

(b) Additional lines based on reduced RNP may also be established to meet operational requirements, e.g., to achieve track-to-airspace or track-to- track separation when these minima reductions cannot be achieved.

(c) When applying paragraph 4-1-1.c(2)(a) or 4-1-1.c(2)(b), a default line must be still be published.

(3) A line based on reduced FAS RNP may be established to achieve minimum DA to FROP distance. A default line is not required in this case.

4-1-2. Feeder segment. Construct feeder segments as described in chapter 2 using one or more TF or RF legs. Segment XTT is RNAV 1 STAR/Feeder/Arrival flight phase. Secondary areas apply.

4-1-3. Initial segment. Construct initial segments as described in chapter 1 using one or more TF or RF legs. Segment XTT is RNP AR APCH, Initial flight phase. Secondary areas do not apply.

4-1-4. Intermediate segment. Construct intermediate segments as described in chapter 1 using one or more TF or RF legs. Segment XTT is RNP AR APCH, Intermediate flight phase. Secondary areas do not apply. FB turns at the PFAF are limited to a maximum of 15 degrees. See appendix C for RF transition to ILS.

Section 4-2. Final Approach Segment (FAS)

4-2-1. General. Segment XTT is RNP AR APCH, final flight phase. Secondary areas do not apply. RNP approaches are 3D procedures; the final segment provides the pilot with final segment vertical and lateral path deviation information based on baro-VNAV systems. The minimum HAT value is 250 feet.

a. Vertical path. Chapter 1 applies with the following additional guidance determining a PFAF on an RF leg.

(1) Use formula 4-2-1 and formula 4-2-2 to determine the PFAFs Cartesian coordinates relative to the LTP/FTP. The location of the PFAF may be calculated geodetically from these Cartesian values.

(a) *Step 1*. Determine the flight track distance from LTP/FTP to PFAF using formula 1-3-3.

(b) *Step 2*. Determine the distance from LTP/FTP to the FROP [see paragraph 4-3-5].

(c) *Step 3*. Subtract *Step 2* from *Step 1* to calculate the distance around the arc to the PFAF from the FROP. Use formula 1-2-12 to determine number of degrees of arc; conversely, use formula 1-2-11 to convert degrees of arc to length.

Formula 4-2-1. PFAF on an RF Leg, Cartesian Coordinate "X" Value

$$\mathrm{X} = \mathrm{d_{FROP}} + \mathrm{R} \times \sin(\alpha)$$

Where:
$\mathrm{d_{FROP}}$ = distance from LTP to final rollout point
α = degrees of arc
R = arc radius

Note: This formula works with any unit so long as $\mathrm{d_{FROP}}$ & R share the same unit.

Example:

$$\mathrm{X} = 9420.55 + 19079 \times \sin(98.9)$$
$$\mathrm{X} \approx 28269.84$$

Formula 4-2-2. PFAF on an RF Leg, Cartesian Coordinate "Y" Value

$$Y = R - [R \times \cos(\alpha)]$$

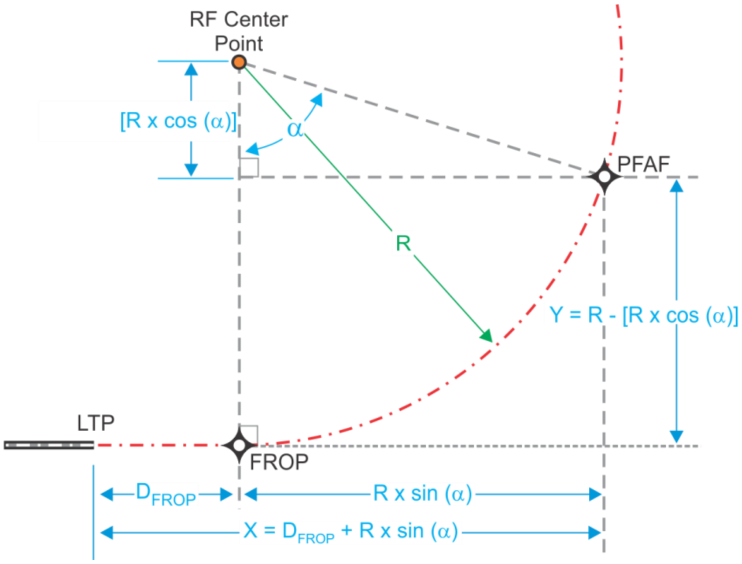
Where:
α = degrees of arc
R = arc radius

Note: This formula produces a value with the same unit as R.

Example:

$Y = 19079 - [19079 \times \cos(98.9)]$
$Y \approx 22030.72$

Figure 4-2-1. Calculating Cartesian Coordinates, PFAF on an RF leg



b. Restrictions. The true vertical path provided by baro-VNAV is influenced by temperature variations; i.e., during periods of cold temperature, the effective glidepath may be lower than published and during periods of hot weather, the effective glidepath may be higher than published. Because of this phenomenon, minimum and maximum temperature limits (for aircraft that are not equipped with temperature compensating systems) are published on the approach chart. Additionally, RNP AR approach procedures are not authorized where a remote altimeter setting source (RASS) is in use or where the final segment overlies precipitous terrain. See paragraph 3-3-4.a(1) to determine the critical temperatures and ΔISA_{LOW} for VEB calculations.

4-2-2. Alignment.

a. The optimum alignment is a TF leg straight in from PFAF to LTP on runway centerline extended. If necessary, the TF course may be offset by up to three degrees. Where the course is offset, it must cross runway centerline extended at least 1500 feet out from LTP.

b. Turns in the FAS and final rollout point (FROP). TF-TF turns are not allowed in the FAS. Where turns are necessary in the FAS an RF leg must be established. The RF leg must terminate and be followed by a TF leg meeting FAS alignment at or prior to reaching the FROP. The *minimum* FROP distance is the greater of the point on the vertical path 500 feet above LTP/FTP elevation or the point in the FAS at distance equal to 15 or 50 seconds prior to the DA point. The number of seconds applied is dependent on the initial missed approach RNP value (RNP_{IMAS}) [see formula 4-2-3]. If an RF leg is the last leg in the intermediate segment, the PFAF must meet the minimum FROP distance.

Formula 4-2-3. Minimum FROP distance

$$\mathrm{d}_{500} = \frac{500 - \mathrm{TCH}}{\tan(\theta)}$$

$$\mathrm{d}_{\mathrm{time}} = \frac{\mathrm{DA} - (\mathrm{LTP}_{\mathrm{elev}} + \mathrm{TCH})}{\tan(\theta)} + \frac{(\mathrm{V}_{\mathrm{KTAS}} + 15) \times 1852}{0.3048} \times \frac{\mathrm{S}}{3600}$$

$\mathrm{d}_{\mathrm{FROP}}$ = greater of $\mathrm{d}_{\mathrm{time}}$ or d_{500}

Where:
$\mathrm{V}_{\mathrm{KTAS}}$ = calculated KTAS using PFAF altitude and final approach KIAS for the fastest CAT
S = seconds of flight

Note: For $\mathrm{RNP}_{\mathrm{IMAS}} \geq 1.0, \mathrm{S} = 15$. For $\mathrm{RNP}_{\mathrm{IMAS}} < 1.0, \mathrm{S} = 50$

Example:

$$\mathrm{d}_{500} = \frac{500 - 47}{\tan(3)}$$

$\mathrm{d}_{500} \approx 8643.75$ feet

$$\mathrm{d}_{\mathrm{time}} = \frac{1760.4 - (1441.6 + 47)}{\tan(3)} + \frac{(152.25 + 15) \times 1852}{0.3048} \times \frac{15}{3600}$$

$\mathrm{d}_{\mathrm{time}} \approx 9420.55$ feet

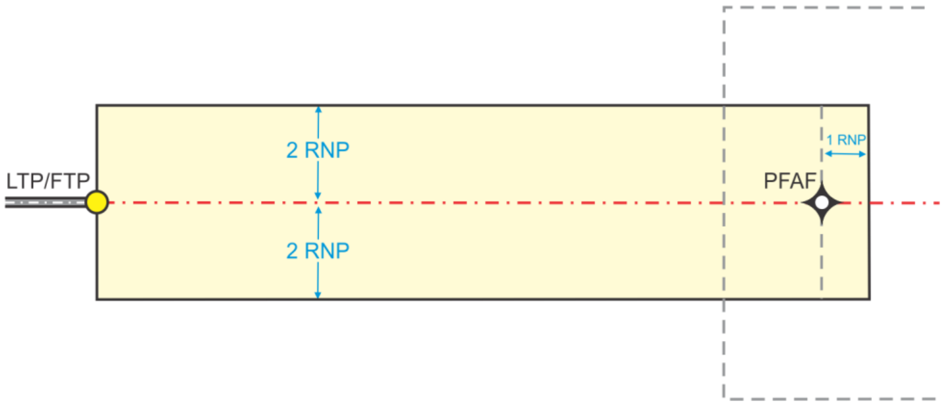
$\mathrm{d}_{\mathrm{FROP}} \approx 9420.55$ feet

4-2-3. Area.

a. Length. The final segment OEA begins 1 × RNP prior to the PFAF and extends to the LTP/FTP.

b. Width. The final segment OEA width is 2 × RNP [see figure 4-2-2].

Figure 4-2-2. Final Segment OEA



4-2-4. Obstacle clearance. An OCS based on the Vertical Error Budget (VEB) is applied. The VEB origin varies depending on whether the procedure is designed for aircraft with wingspans ≤ 262 feet (wide body) or aircraft with wingspans ≤ 136 feet (narrow body). Design for wide body aircraft by default. When the DA can be reduced by at least 50 feet or visibility reduced by ¼ mile or where wide body operations are not anticipated (e.g., where the airport/runway infrastructure cannot support wide body aircraft) the approach may be designed for narrow body aircraft. In either case, the procedure must be properly annotated [see Order 8260.19, paragraph 4-6-10]. The VEB origin also varies depending on whether evaluating a TF or RF leg (with corresponding changes to the OCS height when both leg types are in the FAS).

a. Calculating the VEB. Total VEB is calculated by adding bias errors to four thirds times the standard deviation variations combined via the root sum square method. This is completed for both the PFAF and 250 feet above TDZE to determine the OCS slope and origin.

(1) Bias errors are:

(a) Body geometry error accounts for the low point of the aircraft below the altimeter reference point. This may be the landing gear for aircraft on straight legs, but it could be a wing tip for aircraft on RF legs.

1. Narrow body (BG_{NB}). Wingspan less than or equal to 136 feet.

a Straight legs: fixed at 15 feet

b RF legs: greater of 15 feet or $\frac{136}{2} \times \sin(\phi)$

2. Wide body (BG_{WB}). Wingspan less than or equal to 262 feet.

a Straight legs: fixed at 25 feet

b RF legs: greater of 25 feet or $\frac{262}{2} \times \sin(\phi)$

(b) International standard atmosphere temperature deviation (ISAD) is attributed to temperature induced altimeter error at the specified critical low temperature in relation to the airport ISA (ΔISA_{Low}) [see paragraph 4-2-1.b].

(c) Example bias error calculations.

$$BG_{NB} = \frac{136}{2} \times \sin(\phi)$$
$$BG_{NB} = \frac{136}{2} \times \sin(18)$$
$$BG_{NB} \approx 21.01 \text{ feet}$$

$$ISAD_{PFAF} = \frac{(PFAF_{alt} - LTP_{elev}) \times \Delta ISA_{Low}}{288 + \Delta ISA_{Low} - 0.5 \times 0.00198 \times PFAF_{alt}}$$
$$ISAD_{PFAF} = \frac{(4500 - 1200) \times -20}{288 - 20 - 0.5 \times 0.00198 \times 4500}$$
$$ISAD_{PFAF} \approx -250.43 \text{ feet}$$

$$ISAD_{250} = \frac{(TDZE + 250 - LTP_{elev}) \times \Delta ISA_{Low}}{288 + \Delta ISA_{Low} - 0.5 \times 0.00198 \times (TDZE + 250)}$$
$$ISAD_{250} = \frac{(1202 + 250 - 1200) \times -20}{288 - 20 - 0.5 \times 0.00198 \times (1202 + 250)}$$
$$ISAD_{250} \approx -18.91 \text{ feet}$$

(2) Standard deviation variations are:

(a) Actual navigation performance error (ANPE). An aircraft is expected to be within a radius equal to the required navigation performance (RNP) for the leg 95% of the time. This additive accounts for the 5th percentile that may be operating outside of RNP.

(b) Waypoint precision error (WPR) contains the horizontal position error associated with the coordinate resolution stored in the database and flight management system (FMS) computational accuracy.

(c) Flight technical error (FTE), fixed at 75 feet, is a measure of the pilot and/or autopilot's ability to track the intended flight path.

(d) Altimetry system error (ASE) is the difference between the pressure altitude displayed to the pilot when referencing ISA and the free stream pressure altitude.

(e) Vertical angle error (VAE) relates to the ability of the FMS to determine incremental height above the reference point using distance and intended flight path angle.

(f) Automatic terminal information system (ATIS), fixed at 20 feet, is a conservative assumption that the reported altimeter setting is within 0.02 inches of mercury of the actual pressure value.

(g) Example standard deviation calculations.

$$\mathrm{ANPE} = 1.225 \times \mathrm{RNP} \times \frac{1852}{0.3048} \times \tan(\theta)$$
$$\mathrm{ANPE} = 1.225 \times 0.14 \times \frac{1852}{0.3048} \times \tan(3)$$
$$\mathrm{ANPE} \approx 54.61 \text{ feet}$$

$$\mathrm{WPR} = 60 \times \tan(\theta)$$
$$\mathrm{WPR} = 60 \times \tan(3)$$
$$\mathrm{WPR} \approx 3.14 \text{ feet}$$

$$\mathrm{FTE} = 75 \text{ feet}$$

$$\mathrm{ASE_{PFAF}} = -8.8 \times 10^{-8} \times \mathrm{PFAF_{alt}}^2 + 6.5 \times 10^{-3} \times \mathrm{PFAF_{alt}} + 50$$
$$\mathrm{ASE_{PFAF}} = -8.8 \times 10^{-8} \times 4500^2 + 6.5 \times 10^{-3} \times 4500 + 50$$
$$\mathrm{ASE_{PFAF}} \approx 77.47 \text{ feet}$$

$$\mathrm{ASE_{250}} = -8.8 \times 10^{-8} \times (\mathrm{TDZE} + 250)^2 + 6.5 \times 10^{-3} \times (\mathrm{TDZE} + 250) + 50$$
$$\mathrm{ASE_{250}} = -8.8 \times 10^{-8} \times (1202 + 250)^2 + 6.5 \times 10^{-3} \times (1202 + 250) + 50$$
$$\mathrm{ASE_{250}} \approx 59.25 \text{ feet}$$

$$\mathrm{VAE_{PFAF}} = \frac{\mathrm{PFAF_{alt}} - \mathrm{LTP_{elev}}}{\tan(\theta)} \times (\tan(\theta) - \tan(\theta - 0.01))$$
$$\mathrm{VAE_{PFAF}} = \frac{4500 - 1200}{\tan(3)} \times (\tan(3) - \tan(3 - 0.01))$$
$$\mathrm{VAE_{PFAF}} \approx 11.02 \text{ feet}$$

$$\mathrm{VAE_{250}} = \frac{\mathrm{TDZE} + 250 - \mathrm{LTP_{elev}}}{\tan(\theta)} \times (\tan(\theta) - \tan(\theta - 0.01))$$
$$\mathrm{VAE_{250}} = \frac{1202 + 250 - 1200}{\tan(3)} \times (\tan(3) - \tan(3 - 0.01))$$
$$\mathrm{VAE_{250}} \approx 0.84 \text{ feet}$$

$$\mathrm{ATIS} = 20 \text{ feet}$$

(3) Example VEB calculations.

$$\mathrm{VEB} = \mathrm{BG} - \mathrm{ISAD} + \frac{4}{3}\sqrt{\mathrm{ANPE}^2 + \mathrm{WPR}^2 + \mathrm{FTE}^2 + \mathrm{ASE}^2 + \mathrm{VAE}^2 + \mathrm{ATIS}^2}$$
$$\mathrm{VEB_{PFAF}} = 21.01 + 250.43 + \frac{4}{3}\sqrt{54.61^2 + 3.14^2 + 75^2 + 77.47^2 + 11.02^2 + 20^2}$$
$$\mathrm{VEB_{PFAF}} \approx 435.50 \text{ feet}$$
$$\mathrm{VEB_{250}} = 21.01 + 18.91 + \frac{4}{3}\sqrt{54.61^2 + 3.14^2 + 75^2 + 59.25^2 + 0.84^2 + 20^2}$$
$$\mathrm{VEB_{250}} \approx 189.16 \text{ feet}$$

b. Calculating the OCS slope. The OCS slope is calculated using formula 4-2-4.

Formula 4-2-4. VEB OCS Slope

$$\mathrm{OCS_{slope}} = \frac{(\mathrm{PFAF_{alt}} - \mathrm{TDZE} - 250) \times \tan(\theta)^{-1}}{\mathrm{PFAF_{alt}} - \mathrm{TDZE} - \mathrm{VEB_{PFAF}} - 250 + \mathrm{VEB_{250}}}$$

Where:
$\mathrm{VEB_{PFAF}}$ = amount of vertical error budget (feet) at the PFAF
$\mathrm{VEB_{250}}$ = amount of vertical error budget (feet) at a point 250 feet above TDZE

Example:

$$\mathrm{OCS_{slope}} = \frac{(4500 - 1202 - 250) \times \tan(3)^{-1}}{4500 - 1202 - 435.50 - 250 + 189.16}$$
$$\mathrm{OCS_{slope}} \approx 20.76$$

c. Calculating the OCS LTP/FTP to origin distance ($\mathrm{d_{VEB}}$). The $\mathrm{d_{VEB}}$ is calculated with formula 4-2-5 by subtracting the distance required for the OCS to reach $\mathrm{VEB_{250}}$ below glidepath from the along glidepath distance projected from the threshold to a point 250 feet above TDZE.

Formula 4-2-5. VEB OCS Origin

$$\mathrm{d_{VEB}} = \frac{\mathrm{TDZE} + 250 - \mathrm{LTP_{elev}} - \mathrm{TCH}}{\tan(\theta)} - (\mathrm{TDZE} + 250 - \mathrm{VEB_{250}} - \mathrm{LTP_{elev}}) \times \mathrm{OCS_{slope}}$$

Where:
$\mathrm{VEB_{250}}$ = amount of vertical error budget (feet) at a point 250 feet above TDZE

Example:

$$\mathrm{d_{VEB}} = \frac{1202 + 250 - 1200 - 55}{\tan(3)} - (1202 + 250 - 189.16 - 1200) \times 20.76$$
$$\mathrm{d_{VEB}} \approx 2454.43 \text{ feet}$$

Formula 4-2-6. VEB OCS Elevation

$$\mathrm{OCS_{elev}} = \mathrm{LTP_{elev}} + \frac{\mathrm{d_{LTP}} - \mathrm{d_{VEB}}}{\mathrm{OCS_{slope}}}$$

Where:
$\mathrm{d_{LTP}}$ = along-track distance (feet) from LTP to the point of interest
$\mathrm{d_{VEB}}$ = distance (feet) from LTP to VEB OCS origin as per leg type

Example:

$$\mathrm{OCS_{elev}} = 1125.4 + \frac{19747.38 - 4063.28}{21.67}$$
$$\mathrm{OCS_{elev}} \approx 1849.17 \text{ feet}$$

d. Obstacle evaluation. Use formula 4-2-6 to determine the VEB OCS elevation. Apply the OCS applicable to the leg type to the entire leg [see paragraph 1-2-5.b(1)(b) and figure 4-2-3]. Obstacles that penetrate an OCS may be mitigated by one of the following:

(1) Reduce the RNP value for the leg or change the lateral path.

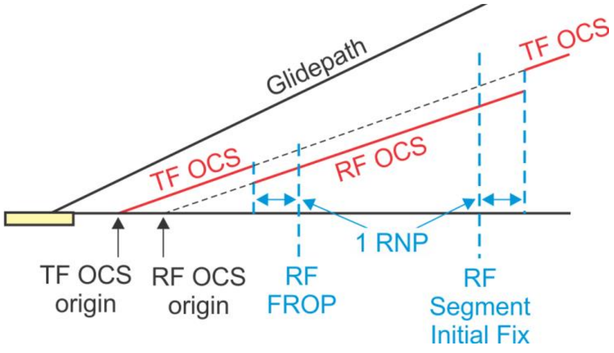
(2) Raise GPA.

(3) Raise TCH.

(4) Adjust the DA [see formula 4-2-7].

Note: The VEB slope and origin is impacted by changes to RNP, GPA, and/or TCH. When paragraphs 4-2-4.d(1), 4-2-4.d(2), and/or 4-2-4.d(3) are applied, the VEB must be recalculated and the OCS re-evaluated.

Figure 4-2-3. Final Segment OCS Elevation



Formula 4-2-7. DA Distance Based on VEB OCS Penetration

$$d_{DA} = d_{LTP} + p \times OCS_{slope}$$

Where:
p = penetration (feet)of the OCS
d_{LTP} = along track distance (feet) from the LTP/FTP to obstacle

Example:

$d_{DA} = 4514.5 + 8.3 \times 23.42$
$d_{DA} \approx 4708.89$ feet

e. Published decision altitude. If the FAS OCS is not penetrated, the minimum HAT value of 250 feet applies. Except when using an RNAV MAS [see paragraph 4-3-1.a(2)], the DA distance from LTP/FTP must also not be less than missed approach $d_{heightloss}$ [see formula 4-3-4] plus d_{VEB}.

Section 4-3. Missed Approach Segment (MAS)

4-3-1. General. Segment XTT is RNP AR APCH, missed approach flight phase. Secondary areas do not apply. Optional construction RNAV 1 segment RNP, missed approach flight phase with secondary areas.

a. MAS RNP. Use of RNP < 1.0 in the missed approach limits may limit availability of the procedure. In order to serve the largest number of users, specify a default or RNAV MAS where possible.

(1) Default MAS (RNP AR APCH with RNP 1.0). Use except where not operationally beneficial. The construction is a continuation of the FAC. The OEA expands from the FAS RNP to an RNP value of 1.0.

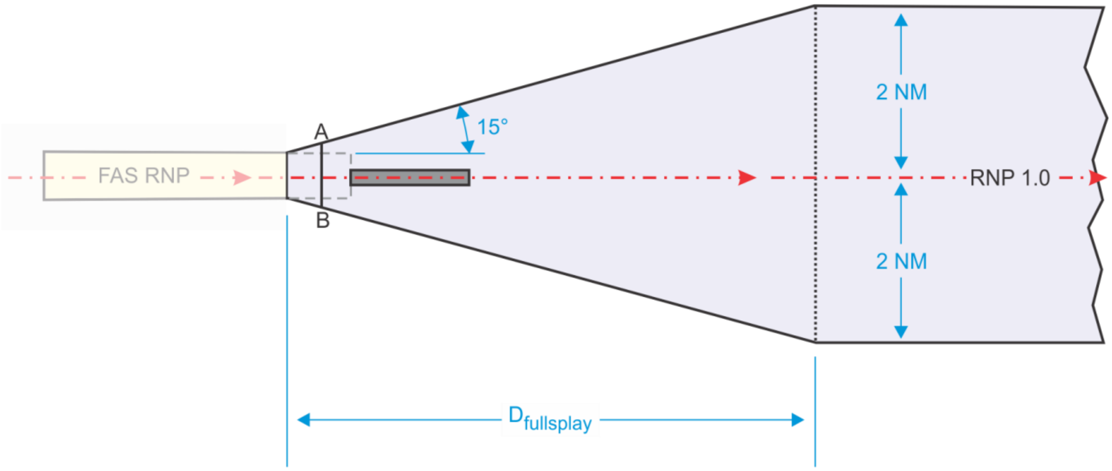
(2) RNAV MAS (RNAV 1 missed). Use where operationally beneficial (i.e., turn-at-altitude construction to allow a turn earlier than permitted by default MAS construction).

(3) Reduced RNP MAS (RNP AR APCH with RNP <1.0). Use when previous options are not practical. Construct straight or turning (using RF legs) missed approach.

4-3-2. Default MAS OEA construction. Establish the MAS using TF and/or RF legs.

a. Straight. The MAS leg expands from the FAS OEA DA at an angle of at least 15 degrees relative to course centerline until reaching a width of ± 2 NM (RNP 1.0) [see figure 4-3-1]. RNP 1.0 width must be achieved at or prior to reaching the first missed approach fix or the clearance limit if no other fix exists beyond the DA. The along-track distance (NM) required to complete the splay ($D_{fullsplay}$) may be calculated using formula 4-3-1 When RNP 1.0 is not reached using a 15-degree splay, establish an RNAV MAS or splay at the angle required to reach ± 2 NM at the turn point/fix (S_{angle}).

Figure 4-3-1. Default MAS OEA, Straight



Formula 4-3-1. Along-Track Distance to Complete Splay

$$\mathrm{D_{fullsplay}} = \frac{2 \times (1 - \mathrm{RNP_{FAS}})}{\tan(\mathrm{S_{angle}})}$$

Where:
$\mathrm{RNP_{FAS}}$ = RNP value (NM) of the final segment
$\mathrm{S_{angle}}$ = missed approach OEA splay angle relative to the course centerline

Example:

$$\mathrm{D_{fullsplay}} = \frac{2 \times (1 - 0.3)}{\tan(15)}$$
$$\mathrm{D_{fullsplay}} \approx 5.22\ \mathrm{NM}$$

b. Turning missed approach. Apply paragraph 4-3-2.a in conjunction with the following paragraphs.

(1) TF-TF connection. Use paragraph 1-2-5.d(1) in cases where RNP 1.0 width is achieved at a sufficient distance prior to the turn fix to permit construction. Otherwise, construct as follows:

(a) *Step 1*. Construct the leg following the turn fix (outbound leg) at full width.

(b) *Step 2*. Determine the DTA using formula 1-2-1. Establish point EB at the turn-side boundary of the inbound leg abeam the DTA. Establish point EE at the turn-side boundary of the outbound leg abeam the DTA.

(c) *Step 3*.

1 Case 1 [see figure 4-3-2]. Point EB is internal to the outbound leg OEA. In this case, no expansion is required. The area is a simple connection of the inbound and outbound leg boundaries.

2 Case 2 [see figure 4-3-3]. Point EB is external to the outbound leg OEA. In this case, expand the boundaries with a line connecting points EB and EE.

(2) TF-RF and RF-RF connections. Paragraph 1-2-5.d(3) applies.

Figure 4-3-2. Default MAS OEA Construction, Turning TF-TF Connection, Splay Greater than 15 degrees, Case 1

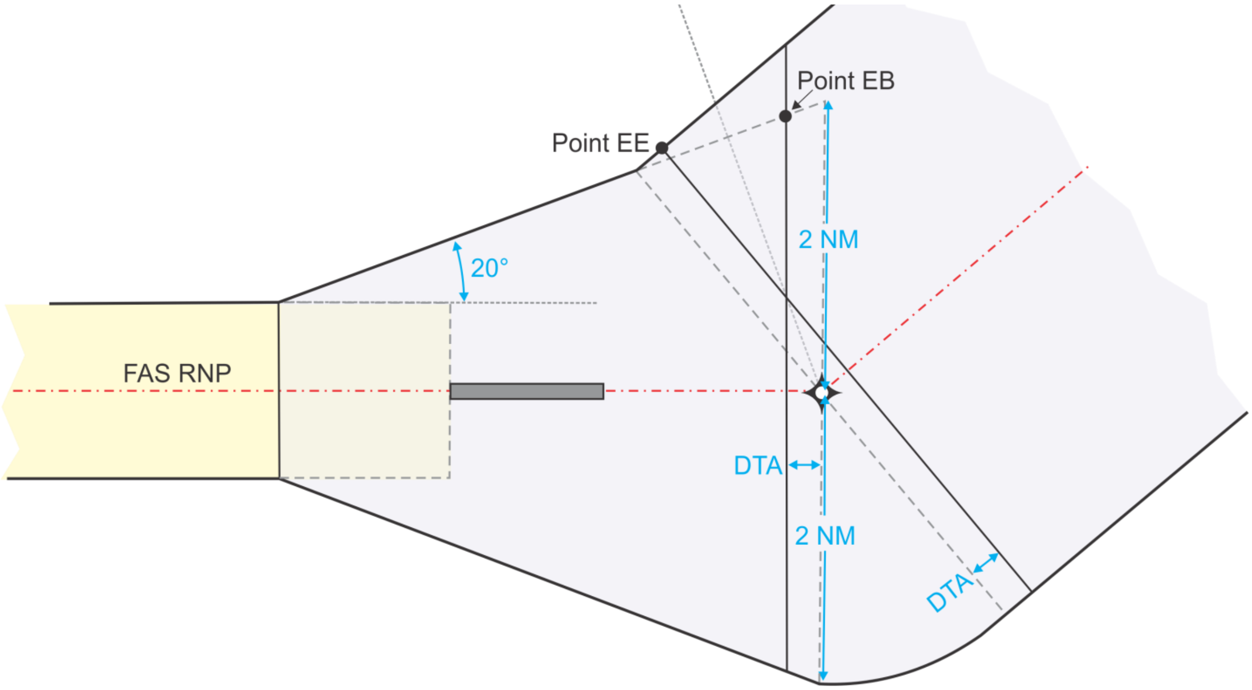
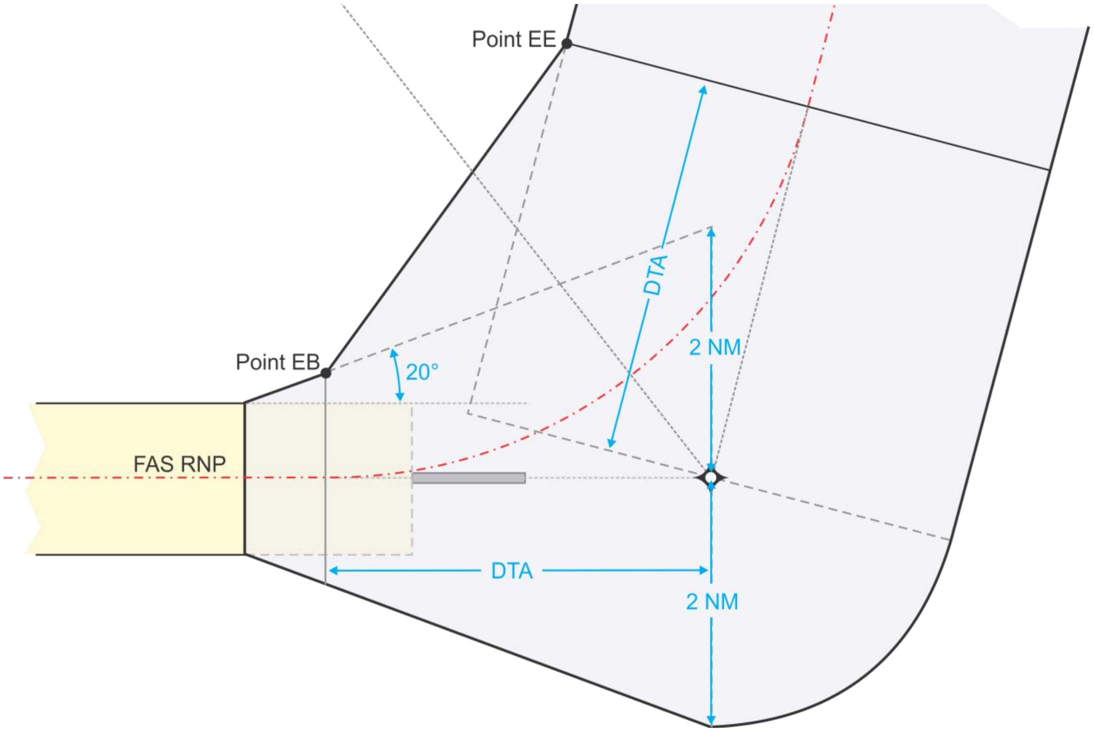


Figure 4-3-3. Default MAS OEA Construction, Turning TF-TF Connection, Splay Greater than 15 degrees, Case 2



4-3-3. RNAV MAS OEA construction [see figure 4-3-4]. Use LNAV/VNAV construction in accordance with section 3-5 and section 3-7 with the following exceptions;

a. The RNP AR obstacle evaluation sequence applies [see Order 8260.19, paragraph 2-11-4].

b. Paragraph 3-6-2 applies except the 15-degree OEA splay begins at the outer edge of the final OEA at the DA point instead of the final secondary area (i.e., the secondary start width is zero). Calculate the distance from course centerline to the primary and secondary boundary of the OEA at any distance from line C-D using formula 4-3-2.

Formula 4-3-2. RNP AR RNAV MAS Primary and Secondary Boundary

$$\mathrm{MAS}_{\mathrm{Yprimary}} = \frac{\mathrm{D}_{\mathrm{splay}} \times \tan(15) \times (1 - \mathrm{RNP}_{\mathrm{FAS}})}{1.5 - \mathrm{RNP}_{\mathrm{FAS}}} + 2 \times \mathrm{RNP}_{\mathrm{FAS}}$$

$$\mathrm{MAS}_{\mathrm{Ysecondary}} = \mathrm{D}_{\mathrm{splay}} \times \tan(15) + 2 \times \mathrm{RNP}_{\mathrm{FAS}}$$

Where:
$\mathrm{D}_{\mathrm{splay}}$ = along-track distance (NM) from beginning of MAS splay to point of interest
$\mathrm{RNP}_{\mathrm{FAS}}$ = RNP value (NM) of the final segment

Example:

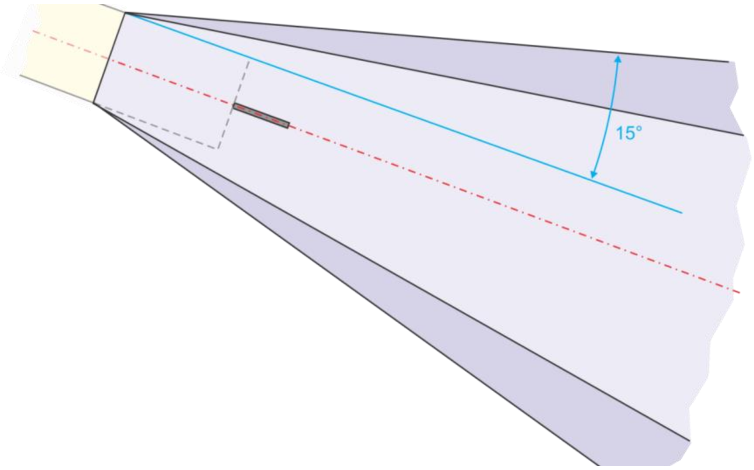
$$\mathrm{MAS}_{\mathrm{Yprimary}} = \frac{4.48 \times \tan(15) \times (1 - 0.3)}{1.5 - 0.3} + (2 \times 0.3)$$

$$\mathrm{MAS}_{\mathrm{Yprimary}} \approx 1.3\ \mathrm{NM}$$

$$\mathrm{MAS}_{\mathrm{Ysecondary}} = 4.48 \times \tan(15) + 2 \times 0.3$$

$$\mathrm{MAS}_{\mathrm{Ysecondary}} \approx 1.8\ \mathrm{NM}$$

Figure 4-3-4. Missed Approach Segment, RNAV MAS Area



4-3-4. Reduced RNP MAS. Establish using TF and/or RF legs. See paragraph 1-2-5.b for OEA construction. Where turns are necessary, the turn must occur after passing 500 feet above airport elevation and where possible, not prior to the DER. At no point may the turn be specified prior to $D_{MASturn}$ from DA [see formula 4-3-3].

Formula 4-3-3. Reduced RNP MAS, Minimum Distance DA to Turn

$$D_{MASturn} = \frac{(V_{KTAS} + 15) \times 10}{3600}$$

Where:
V_{KTAS} calculated KTAS using DA and final segment KIAS for the fastest CAT

Example:

$$D_{MASturn} = \frac{(170.38 + 15) \times 10}{3600}$$
$$D_{MASturn} \approx 0.51 \text{ NM}$$

4-3-5. OCS evaluation. The following applies to default and reduced RNP missed approaches. For RNAV missed approaches see sections 3-5, 3-6, and 3-7

a. General. The OCS is composed of sections 1a and 1b, which are separated by line A-B.

(1) Section 1a. The OCS extends from the DA point downward at the VEB OCS slope ratio for a distance of $d_{heightloss}$ using formula 4-3-4 measured along the final course track to line A-B.

Formula 4-3-4. Height Loss Distance

$$d_{heightloss} = \frac{50}{\tan(\theta)}$$

Example:

$$d_{heightloss} = \frac{50}{\tan(3)}$$
$$d_{heightloss} \approx 954.06 \text{ feet}$$

(2) Section 1b OCS. From line A-B, section 1b OCS rises at a 40:1 slope Calculate the HMAS at line A-B using formula 4-3-5. Obstacle distance is measured as the along-track distance from line A-B to a point abeam the obstruction.

Formula 4-3-5. HMAS at line AB

$$\mathrm{HMAS} = \mathrm{LTP_{elev}} + \frac{\mathrm{d_{DA}} - \mathrm{d_{VEB}} - \mathrm{d_{heightloss}}}{\mathrm{OCS_{slope}}}$$

Where:
$\mathrm{d_{DA}}$ = distance (feet) from LTP to DA
$\mathrm{d_{VEB}}$ = distance (feet) from LTP to VEB OCS origin
$\mathrm{d_{heightloss}}$ = distance (feet) required to lose 50 feet on glidepath

Example:

$$\mathrm{HMAS} = 1200 + \frac{3758.74 - 2454.55 - 954.06}{20.76}$$
$$\mathrm{HMAS} \approx 1216.87 \text{ feet}$$

b. OCS penetrations. Obstacles must not penetrate the OCS. Where obstructions penetrate the OCS, mitigate using one or more of the following actions:

(1) Revise the MAS course.

(2) Specify a MAS climb gradient [see paragraph 1-3-1.g(2)].Calculate the altitude above which the climb gradient is no longer required using formula 4-3-6. This altitude is used in formula 1-3-7 to calculate the required climb gradient.

Note: If the climb gradient exceeds 425 ft/NM, evaluate the MAS using the OCS slope appropriate for 425 ft/NM (18.82:1) and adjust DA for the remaining penetration per paragraph 4-3-5.b(3).

Formula 4-3-6. CG Termination Altitude

$$\mathrm{CG_{term}} = \frac{(\mathrm{O_{MSL}} - \mathrm{OCS_{start}}) \times 8000 \times 0.3048}{1852} + \mathrm{AC_{start}}$$

Where:
$\mathrm{OCS_{start}}$ = starting altitude (feet)of the OCS
$\mathrm{AC_{start}}$ = starting altitude (feet) of the aircraft (e.g., SOC)

Example:

$$\mathrm{CG_{term}} = \frac{(2147.41 - 1217.37) \times 8000 \times 0.3048}{1852} + 1402$$
$$\mathrm{CG_{term}} = 2626.52 \text{ feet}$$

(3) Adjust DA. To determine the DA required for an OCS penetration, calculate the HAT adjustment using formula 4-3-7 and add the result to the original DA. Use formula 1-3-3 to determine adjusted DA distance from the LTP/FTP.

Formula 4-3-7. HAT Adjustment for MAS OCS Penetration

$$\Delta \mathrm{DA} = \frac{\mathrm{p} \times \tan(\theta) \times \mathrm{MA_{slope}} \times \mathrm{FAS_{slope}}}{\mathrm{MA_{slope}} + \mathrm{FAS_{slope}}}$$

Where:
p = penetration (feet) of the MAS OCS
$\mathrm{MA_{slope}}$ = missed approach OCS slope ratio
$\mathrm{FAS_{slope}}$ = final approach OCS slope ratio

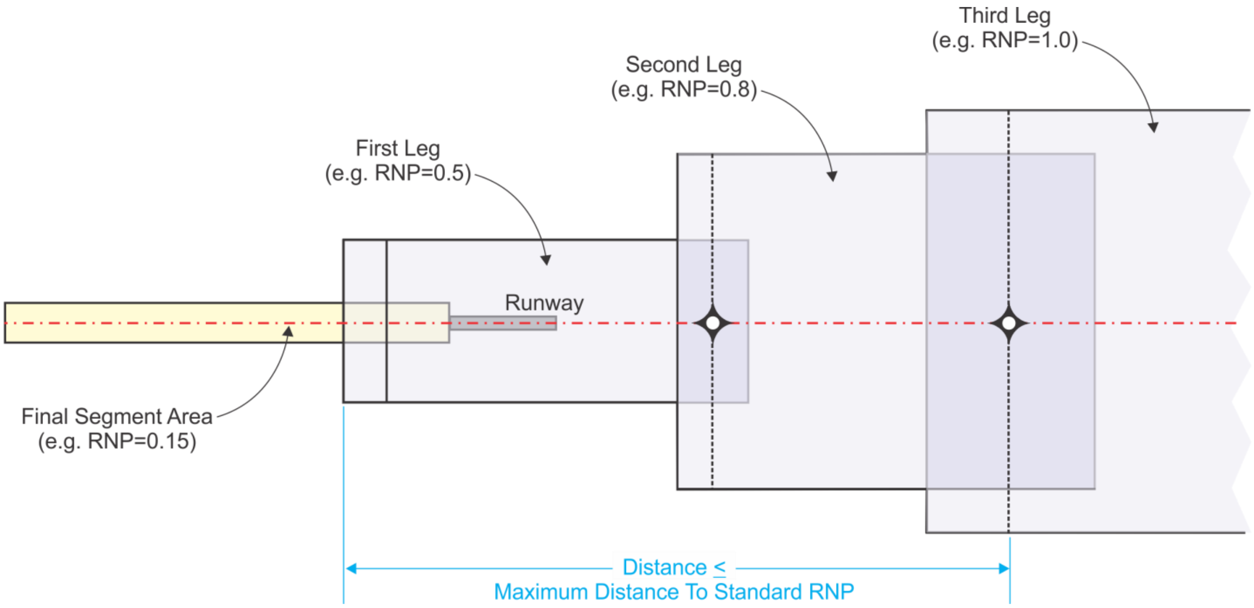
Example:

$$\Delta \mathrm{DA} = \frac{19.3 \times \tan(3) \times 40 \times 23.42}{40 + 23.42}$$

$$\Delta \mathrm{DA} \approx 14.94 \text{ feet}$$

(4) Specify a reduced RNP MAS leg [see paragraph 4-3-1.a(3)].Apply the reduced RNP MAS leg only as far as needed to clear the penetrating obstruction(s) and specify the largest RNP value possible. Calculate the maximum length of the reduced RNP leg using formula 4-3-8 [see figure 4-3-5]. A reduced RNP MAS leg must be a TF or RF leg.

Figure 4-3-5. Maximum Length Reduced RNP MAS



Formula 4-3-8. Max Length Reduced RNP MAS Leg

$$\mathrm{D_{MASRNP}} = (\mathrm{RNP_{MAS}} - 0.05) \times \frac{\mathrm{V_{KTAS}} - 10}{8}$$

Where:
$\mathrm{RNP_{MAS}}$ = RNP value (NM) of the missed approach segment
$\mathrm{V_{KTAS}}$ = calculated KTAS using DA and final approach KIAS for the slowest CAT

Example:

$$\mathrm{D_{MASRNP}} = (0.5 - 0.05) \times \frac{92.93 - 10}{8}$$
$$\mathrm{D_{MASRNP}} \approx 4.66\ \mathrm{NM}$$

THIS PAGE IS INTENTIONALLY LEFT BLANK

Chapter 5. RNAV 1 and RNP 1 Departure Procedures

Section 5-1. General Criteria

5-1-1. General. Segment XTT is RNAV 1 or RNP 1 where RF turns are indicated.

a. ICA. Departure procedures begin with an ICA constructed in accordance with Order 8260.3, paragraph 14-1-6, except where modified by this order.

(1) The ICA must be long enough to allow LNAV engagement (500 feet above airport elevation).

(2) Use formula 1-3-9 to determine ICA length. Where a higher than standard climb gradient terminates prior to the minimum turning altitude, the ICA length will be the distance required to reach the CG termination altitude plus the distance between that altitude and the turning altitude at the reduced CG.

b. Leg type limitations. See Order 8260.46, paragraph 3-1-5 for permissible leg types.

Note: The ‘AER’ in Order 8260.46, table 3-1-1 is for coding purposes. Bearings and distances used for OEA construction and leg length analysis are not measured from the AER.

(1) VA leg. VA legs are all-primary area and the initial departure course is aligned within 15 degrees of the extended runway centerline [see paragraph 5-2-1].

(2) VI leg. VI legs are all-primary area and the initial departure course is aligned with extended runway centerline [see paragraph 5-2-1]. VI legs are associated with CF legs and due to possible Flight Management System route discontinuity, course changes of less than 10 degrees to intercept the CF leg are not authorized without approval from Flight Standards Service.

(3) DF leg. Secondary areas apply once the 15 degree splay from course line (early and late) have reached full primary width. DF-DF is only authorized when the first DF fix is within 15 degrees of the extended runway centerline.

(4) CF leg. Secondary areas apply once the 15 degree splay from course has reached full primary width or as defined in the turning departure section.

c. Leg length. Comply with paragraph 1-2-5.b(1) with the following exceptions.

(1) Do not develop legs exceeding 260 NM to ensure the geodesic path does not exceed the protected airspace for a great circle path.

(2) For LNAV engagement, the first leg must be designed to end at least 500 feet above airport elevation.

(3) To allow a waypoint less than 2 NM from the DER without a climb gradient imposed, a FO fix may be used and published. The ICA must be constructed past the fix to the

LNAV engagement point (500 feet above airport elevation). No turn greater than 15 degrees is permitted at this fix, and a succeeding fix must be established for a DF leg.

(4) The maximum allowable VA or VI leg length is 10 NM.

(5) For FO DF leg length feasibility check see appendix B.

Section 5-2. Straight Departure

5-2-1. Straight departure.

a. VI is a straight departures leg and is aligned with runway centerline extended. VA, DF, CF, and TF are straight departure legs and are aligned within 15 degrees of the runway centerline. Evaluate straight departure legs in accordance with Order 8260.3, paragraph 14-3-1 with the following provisions:

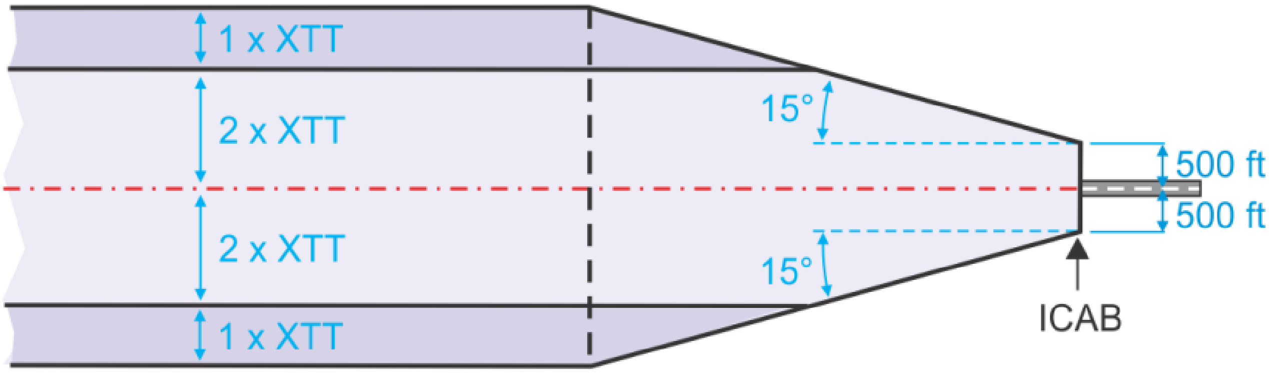
(1) The ICA is aligned along the runway centerline for the distance required for a climb to 500 feet above airport elevation.

(2) Continue splay until reaching basic area width as defined in chapter 1.

(3) The departure may not be offset from runway centerline when the first turn is an RF and the straight segment hasn't reached full OEA width construction [see paragraph 1-2-5.d].

(4) Straight route departure criteria apply only to the first course from DER. Any turn after the first departure course must be evaluated using turning criteria.

Figure 5-2-1. Straight Departure



Section 5-3. Turning Departures

5-3-1. Turning departure.

a. Early and inside turn construction.

(1) First turn. The first turn may be a FB or FO fix or may follow a VA or VI leg.

(a) Turn following VA leg (turn-at-altitude). The turn altitude will either be the minimum required altitude, operationally specified, or determined by obstacle evaluation. The specified turn altitude must equal or exceed the ICA end aircraft altitude.

1. The OEA consists of the ICA, section 1, and section 2. Excluding the ICA, section 1 is defined as the OEA on the DER side of the DRL. Section 2 is the OEA on the SER side of the DRL.

2. ICA. The ICA ends where the specified minimum turning altitude is reached (line L-L'). Construct the ICA from the ICAB to line L-L'. Because ICA length and the OCS elevation at L-L' are dependent on aircraft altitude, establishing an obstacle-based turn altitude may be an iterative process.

3. OEA construction after ICA [see table 5-3-1 and figure 5-3-1].

Table 5-3-1. Early-Turn Tie-Back Points

Turn Angle (measured from RWY C/L ICAE intersection)	Tie-Back Point
≤ 165 degrees	DRP
> 165 degrees	Point P'

a *Step 1.* Construct a line (representing the early-turn flight track) from the tie-back point, to the fix.

b *Step 2.* Construct the inner primary and secondary OEA boundary lines parallel to this line.

c *Step 3.* From the tie-back point, construct a line splaying at 15 degrees to intersect the parallel boundary lines or segment end, whichever occurs earlier. Apply secondary areas only after the 15-degree splay line intersects the primary boundary line. Where the inside turn boundary intersects the ICA inside 15-degree splay line the boundary must be a line connecting the DRP, point A, and full width point or original splay width abeam fix if full width has not been reached [see figure 5-3-2].

Figure 5-3-1. VA-DF Early/Inside Turn Construction after ICA

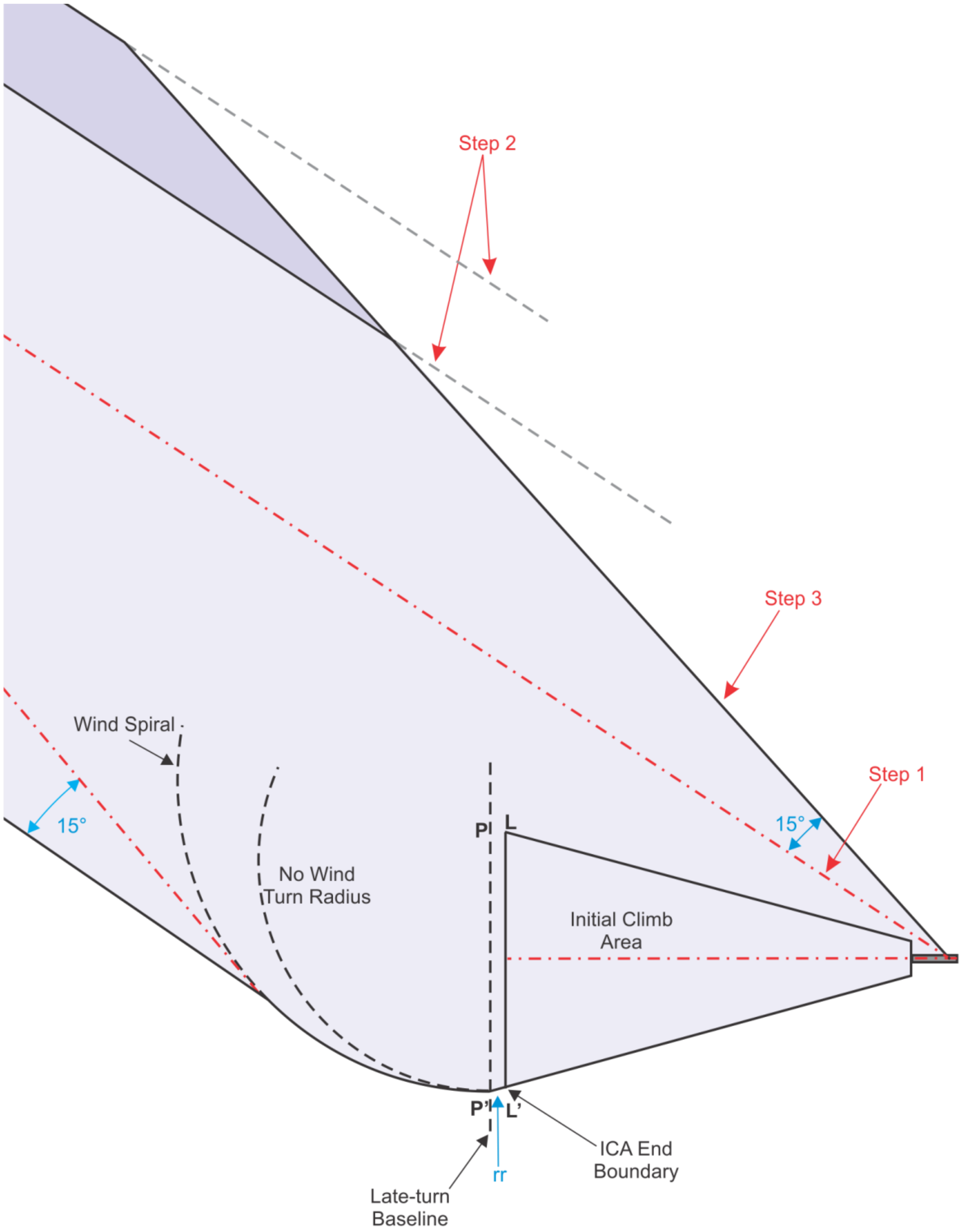
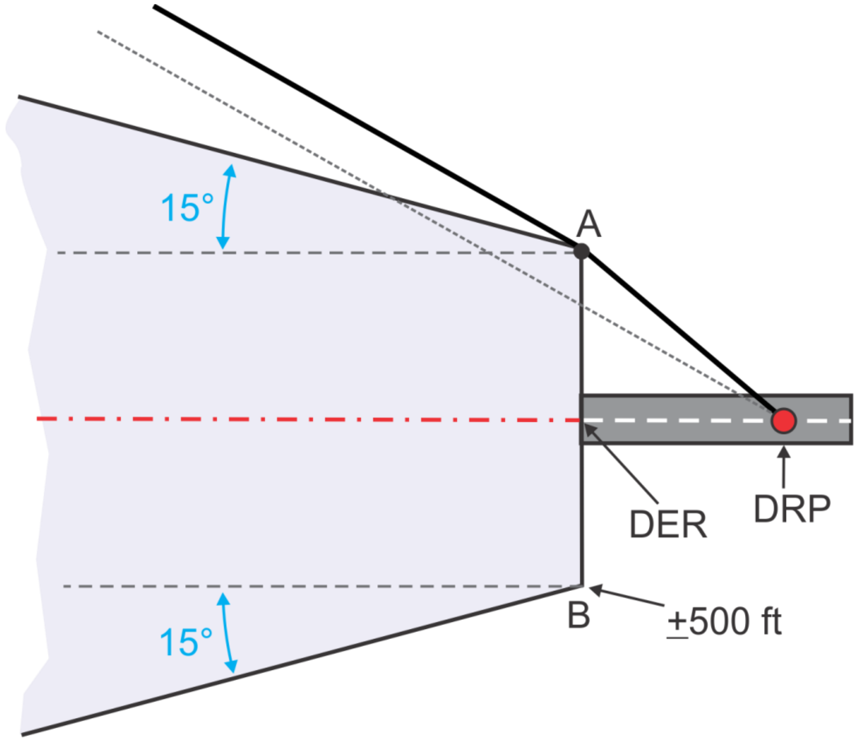


Figure 5-3-2. VA Leg Construction Alternate Inside Boundary



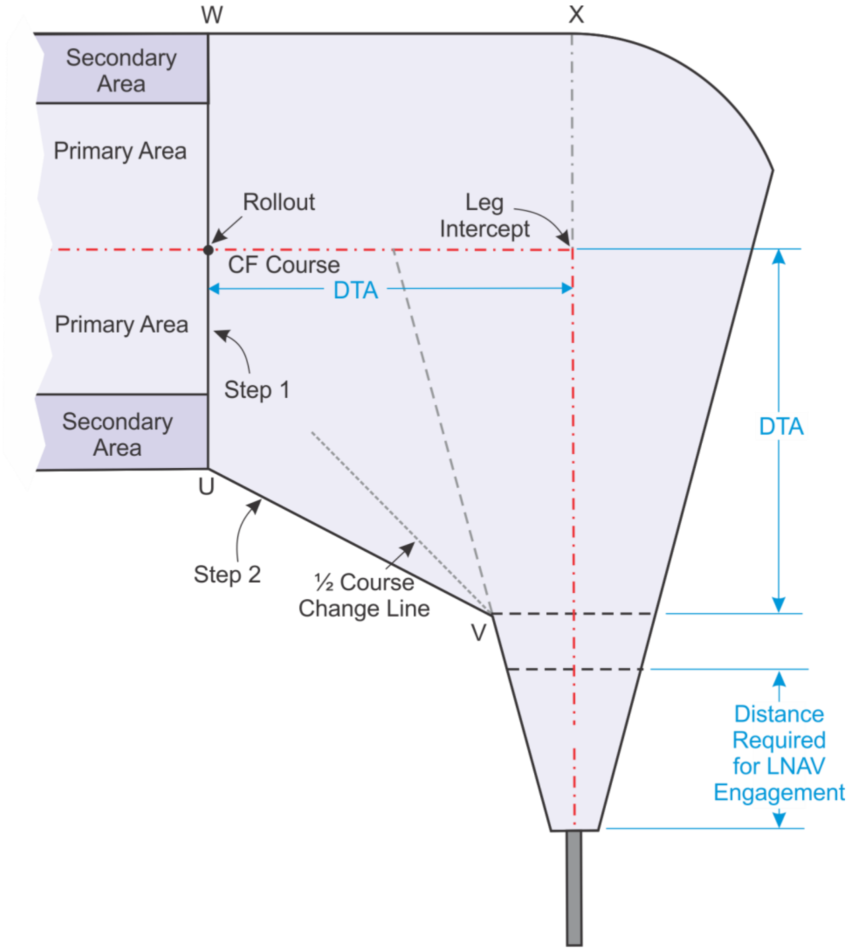
(b) Turn following VI leg. The VI leg contains an ICA constructed in accordance with paragraph 5-1-1.a terminating where LNAV engagement is reached (500 feet above airport elevation). The VI leg OEA is all primary area; secondary area may be applied in the CF leg at the rollout point.

1. OEA construction. Inside expansion starts where the VI DTA area begins (point V) with an angle drawn at one-half of the course change at leg intercept and ends where the angle converges with CF leg secondary boundary. Where the angle does not converge with secondary boundary, draw a line from point V to the inside secondary area boundary abeam the rollout (point U) for area completion [see figure 5-3-3].

a *Step 1.* Along the CF leg course, construct a line perpendicular to the CF leg at a distance from the intercept point equal to the calculated DTA based on the altitude at leg intercept. At the rollout point, the CF leg OEA is the full combined width of the primary and secondary areas.

b *Step 2.* Construct a line (representing the early-turn flight track) from the VI DTA (point V), to the CF leg full width abeam the rollout point.

Figure 5-3-3. Standard VI-CF Construction, 90 Degree Turn



2. Minimum VI-CF leg combination construction. Where a short initial departure leg must be developed, the VI leg length may be designed to the greater of 1 NM from DER or the distance required to climb 500 feet above airport elevation. For this early turn, the OEA is modified to be somewhat like a FB fix turn for inside expansion and additional protection is also provided for outside area expansion built similarly to a FO fix turn.

a For turns of 30 degrees or less, splay 15 degrees relative to the CF leg course from point ICAB and continue this line until intersecting the CF leg secondary area boundary [see Figure 5-3-4]. The secondary area begins where this line crosses the primary area boundary.

b For turns of more than 30 degrees, the inside boundary is from the DRP to the inside secondary area boundary abeam the rollout (point R). From point R, the secondary area tapers 30 degrees inward relative to the CF leg course until the CF leg standard primary area boundary [see figure 5-3-5 thru figure 5-3-7].

c CF leg construction. Along the CF leg course, a rollout point is established from leg intercept at a distance of the calculated DTA, based on the altitude at leg intercept. Establish a full primary and secondary width OEA at the rollout point; the area may or may not be fully utilized based upon the leg intercept turn.

Figure 5-3-4. Minimum VI-CF Construction, Turn 30 Degrees or Less

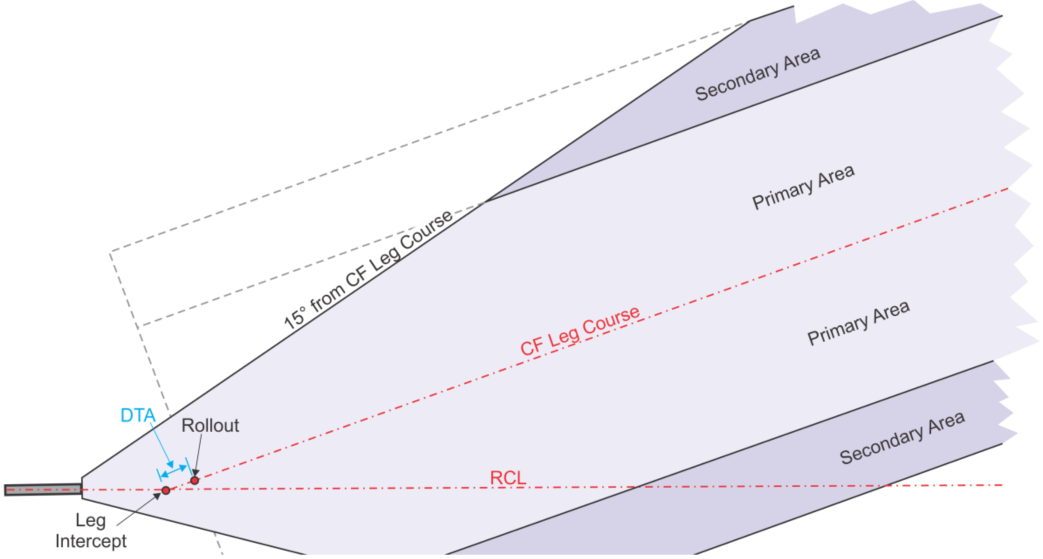


Figure 5-3-5 Minimum VI-CF Construction, Greater than 30-degree Turn

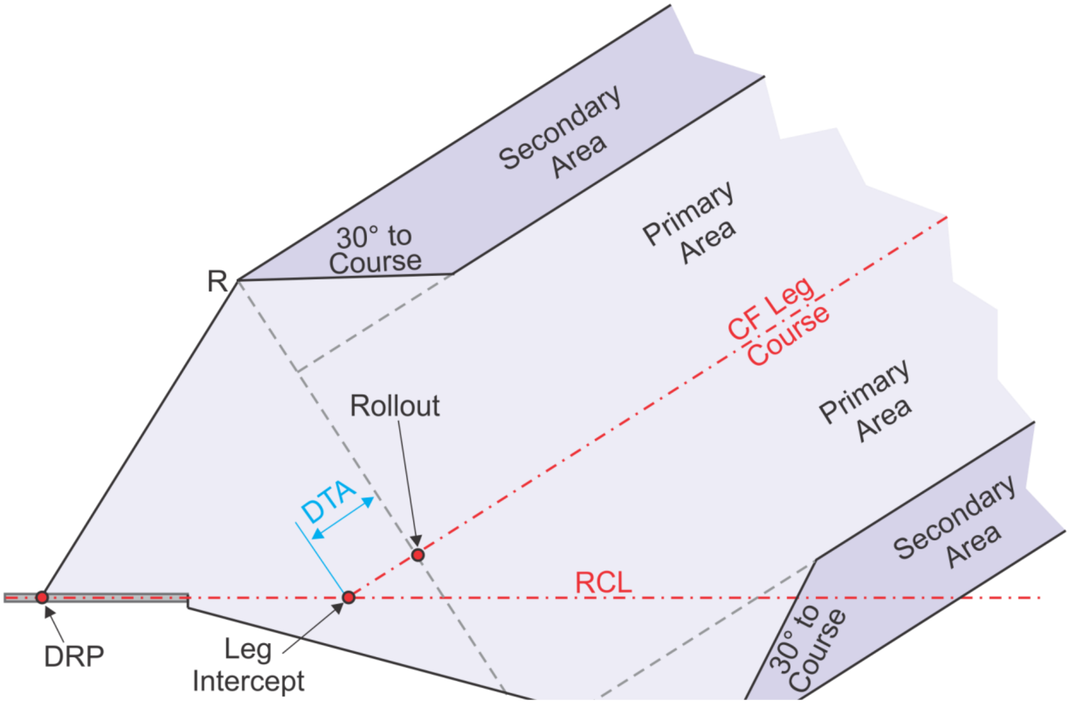


Figure 5-3-6. Minimum VI-CF Construction, 75-degree Turn

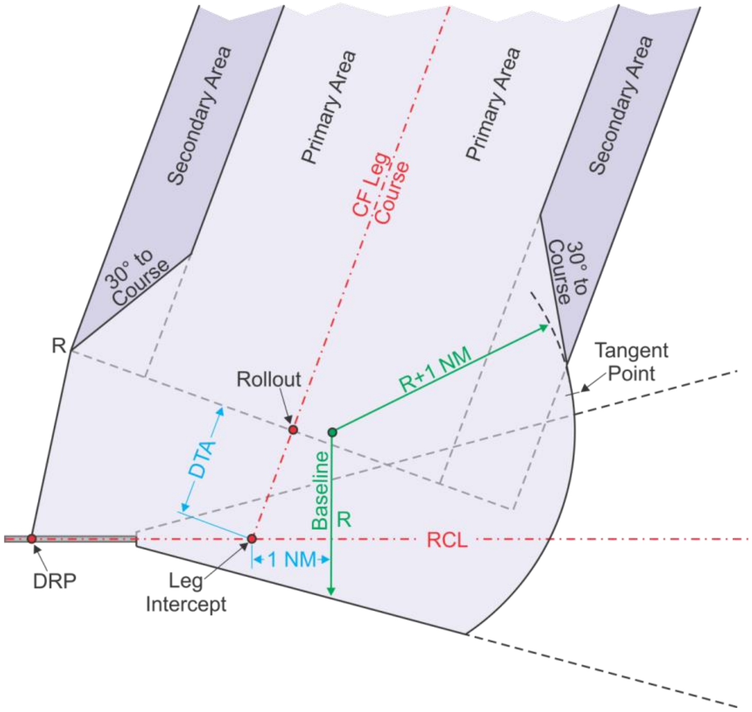
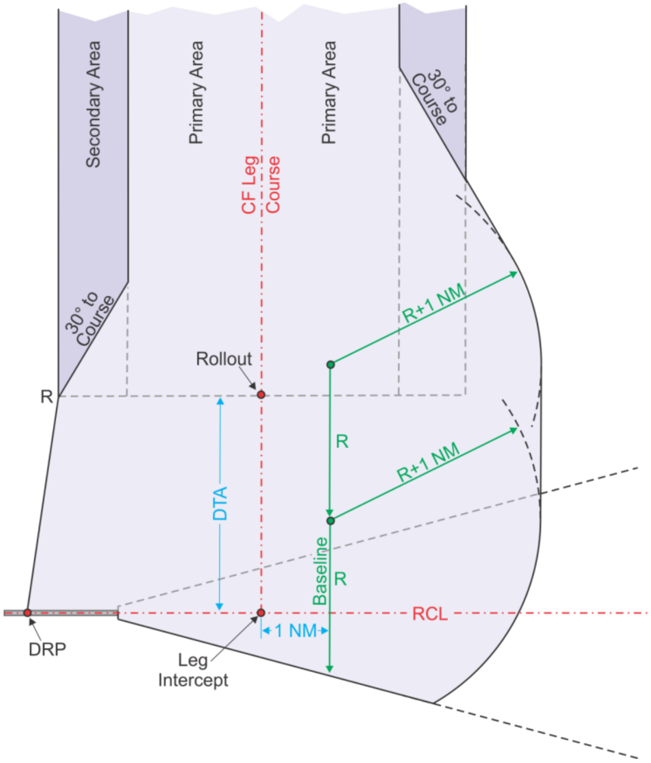


Figure 5-3-7. Minimum VI-CF Construction, 90-degree Turn



(c) Turn-at-fix (FO or FB). Use paragraph 1-2-5.d in cases where full primary and secondary width is achieved at a sufficient distance prior to the turn fix to permit construction. Otherwise, construct as follows:

1. Fix location. The first turn fix must be located on the runway centerline extended (or course line for paragraph 5-2-1.a construction). The distance to the turn fix must be sufficient to result in an early turn baseline at or beyond LNAV engagement (500 feet above airport elevation). Where the first fix must be located at the point the aircraft reaches or exceeds a specific altitude, calculate fix distance from DER with formula 1-3-9 using the standard or assigned CG. If a fix is already established, project the aircraft altitude at the fix in accordance with chapter 1. An RF start point may be located no closer to DER than the greater of 1NM or LNAV engagement (500 feet above airport elevation).

2. Early turn baseline (line L – L’). The early turn baseline is established perpendicular to the inbound leg at the appropriate table 5-3-2 distance prior to the fix.

Table 5-3-2. Early turn Baseline Distance Prior to Turn Fix

Fix Type	Distance
FB Fix	ATT + DTA*
FO Fix	ATT

*DTA = 0 for turns 10 degrees or less

3. Construction points. Inside turn construction uses points PI (primary intersection) and SI (secondary intersection). When defining these points, the leg prior to the turn fix is the inbound leg and the leg following the turn fix is the outbound leg.

a Point SI. This is the connection point for the secondary area boundary and is the intersection of line L-L’ and the inbound leg secondary area boundary. When inbound leg has no secondary, points PI and SI are coincident.

b Point PI. This is the connection point for the primary area boundary and is the intersection of line L-L’ and the inbound leg primary area boundary.

4. Inside turn expansion. Primary/Secondary boundary expansion depends on the location of points PI and SI in relation to the outbound leg OEA boundary lines (extended when necessary) [see table 5-3-3]. Where standard expansion is suitable for one, but not both splays, find the outbound connection point for the non-standard splay abeam the standard connection point. If the primary expansion line would result in a smaller primary area, continue the inbound leg splay until full primary width is reached or until intersecting the angle bisector line whichever occurs first, then splay per table 5-3-3. For RF legs, connect points PI and SI on an early turn baseline L-L’ abeam the RF start point to the outbound leg OEA boundary lines perpendicular to the RF end point [see figure 5-3-11] or point of tangency along the inner primary and secondary boundaries [see figure 5-3-12] whichever occurs first. The inside turn construction line must not result in a splay less than the 15 degree splay of the initial straight leg.

Table 5-3-3. Standard Inside Turn Expansion

Construction Point	External to Outbound leg OEA (both primary and secondary)	Internal to Outbound leg Secondary area	Internal to Outbound leg Primary area
Point SI	Greater of ½ Turn-angle or Inbound leg Splay	15-degrees relative outbound track	
Point PI	Greater of ½ Turn-angle or Inbound leg Splay		15-degrees relative outbound track

5. Construction steps.

a *Step 1.* Construct line L-L' at the appropriate table 5-3-2 distance.

b *Step 2.* Construct secondary area boundary.

1 Case 1. SI is external to the outbound leg OEA. Construct an expansion line from SI to the outbound leg secondary boundary.

2 Case 2. SI is internal to either the outbound leg secondary or primary area. In this case, an alternative construction point (point SI') may be required depending on turn magnitude. Point SI' is the intersection of line L-L' and the outside-turn secondary area boundary. Construct the expansion line from either point SI or SI' to the outbound leg secondary boundary whichever results in the larger area. When the expansion line reaches the outbound secondary boundary before reaching the point of intersection of the inbound and outbound leg secondary boundaries, no expansion is required and the area is a simple connection of the inbound and outbound leg secondary boundaries.

c *Step 3.* Construct the primary area boundary. Construct an expansion line from point PI to the outbound leg primary boundary. When the *Step 2* expansion line reaches the outbound leg primary boundary before reaching point of intersection of the inbound and outbound leg primary boundaries, no expansion is required and the area is a simple connection of the inbound and outbound leg primary boundaries.

Figure 5-3-8. Early/Inside Turn, First Turn, Turn at Fix: SI/PI Collocated and External to Outbound Primary and Secondary

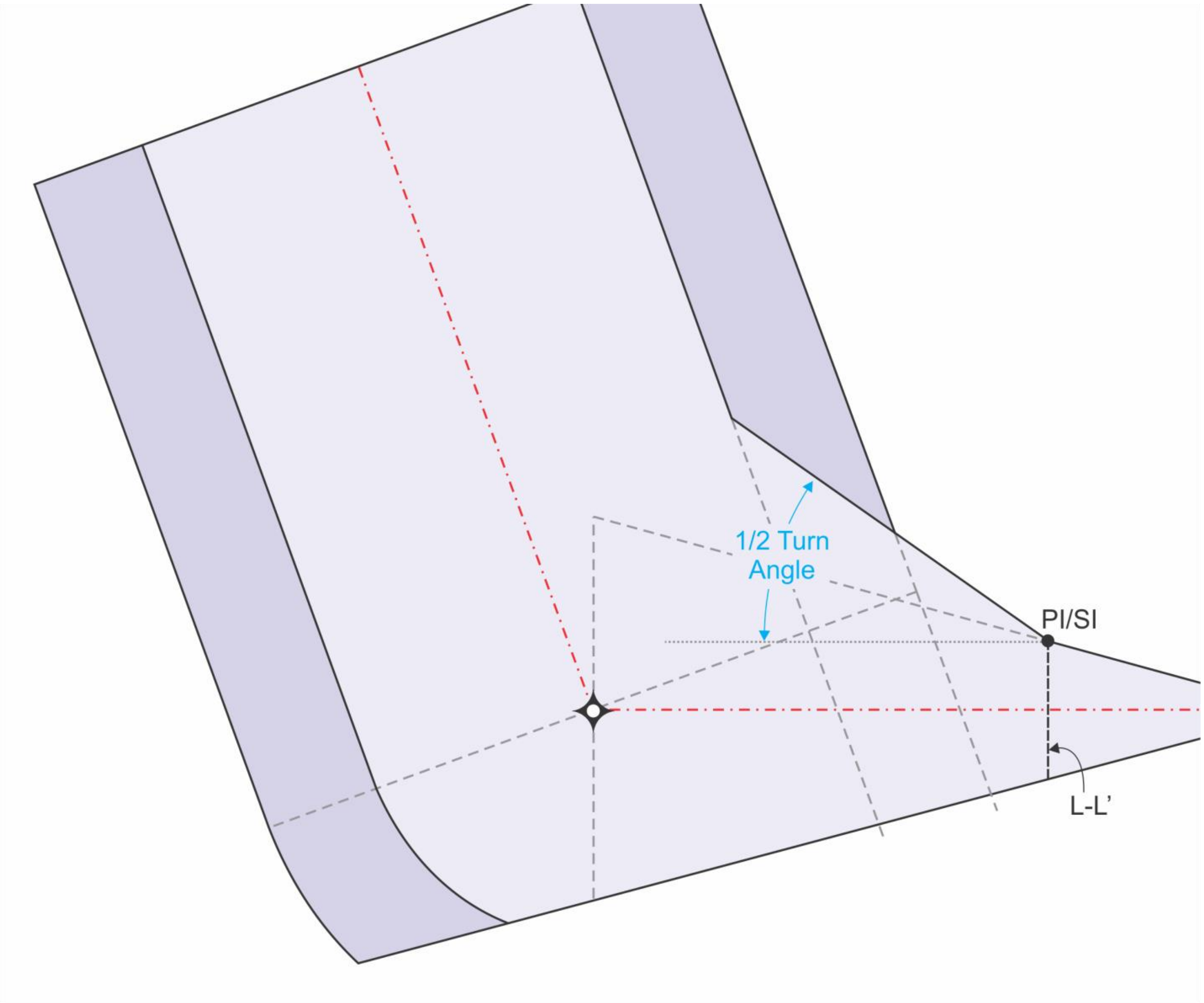


Figure 5-3-9. Early Inside Turn, First Turn,

Turn at Fix: SI/PI internal to Outbound Primary

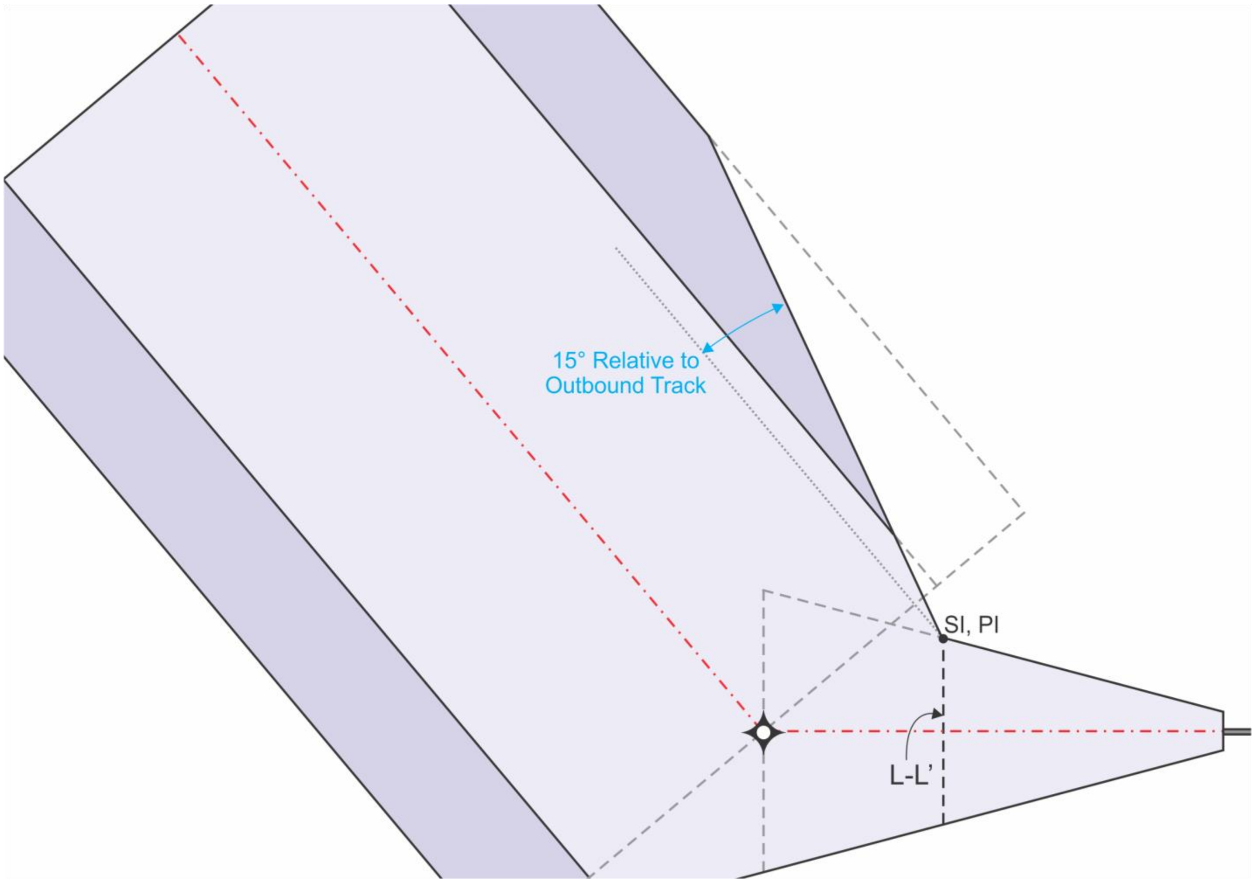


Figure 5-3-10. Early/Inside Turn, First Turn, Turn at Fix:

PI/SI internal to Outbound Secondary, splay from point SI'

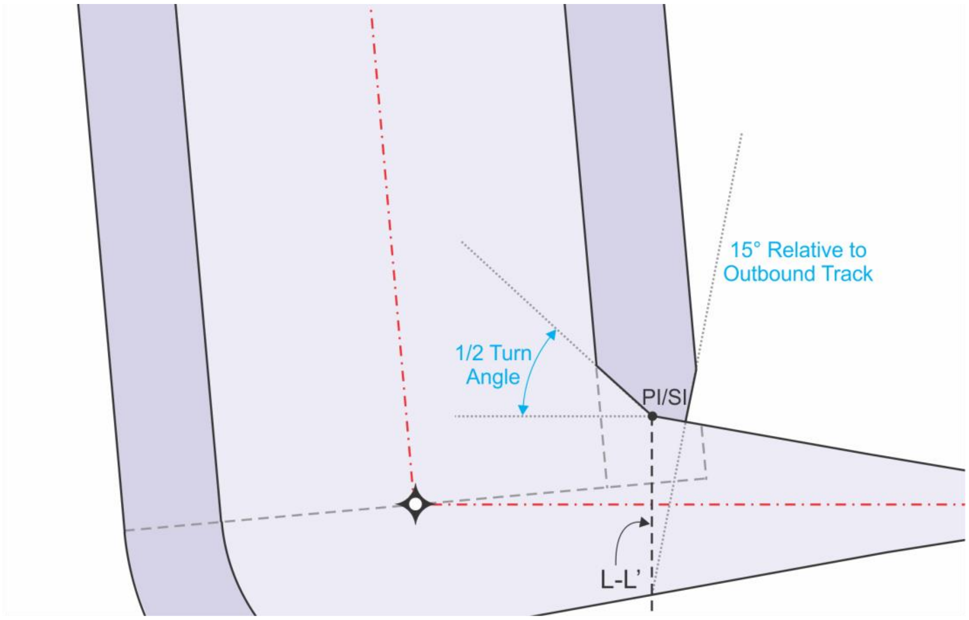


Figure 5-3-11 First Turn, Turn at Fix:

PI/SI connecting to abeam RF end point

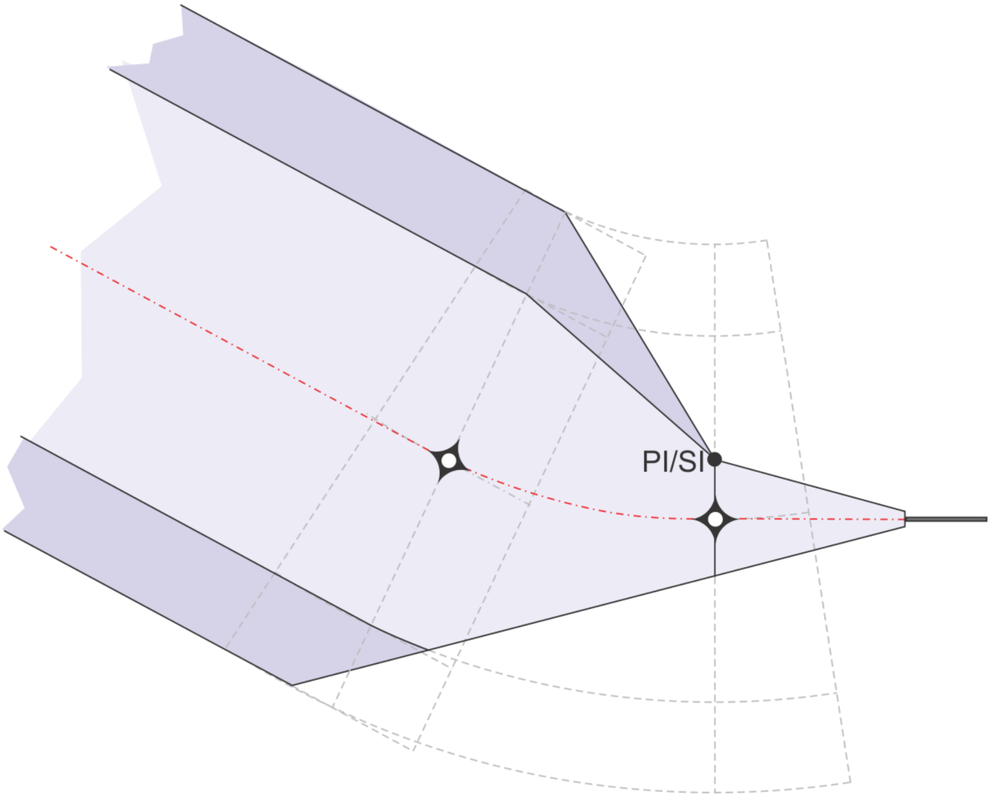
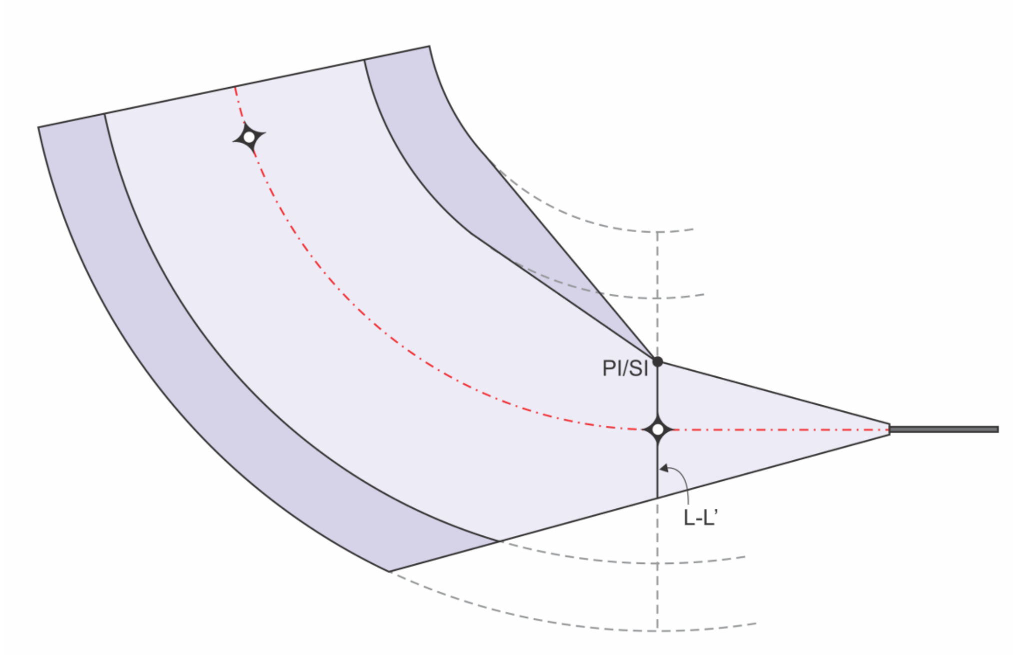


Figure 5-3-12 First Turn, Turn at Fix:

PI/SI connecting to point of tangency prior to RF end point



(2) Second turn. The second turn is specified at either a FB or FO fix.

(a) DF-TF or CF-TF.

1. Construct in accordance with paragraph 5-3-1.a(1)(c) with the following exceptions:

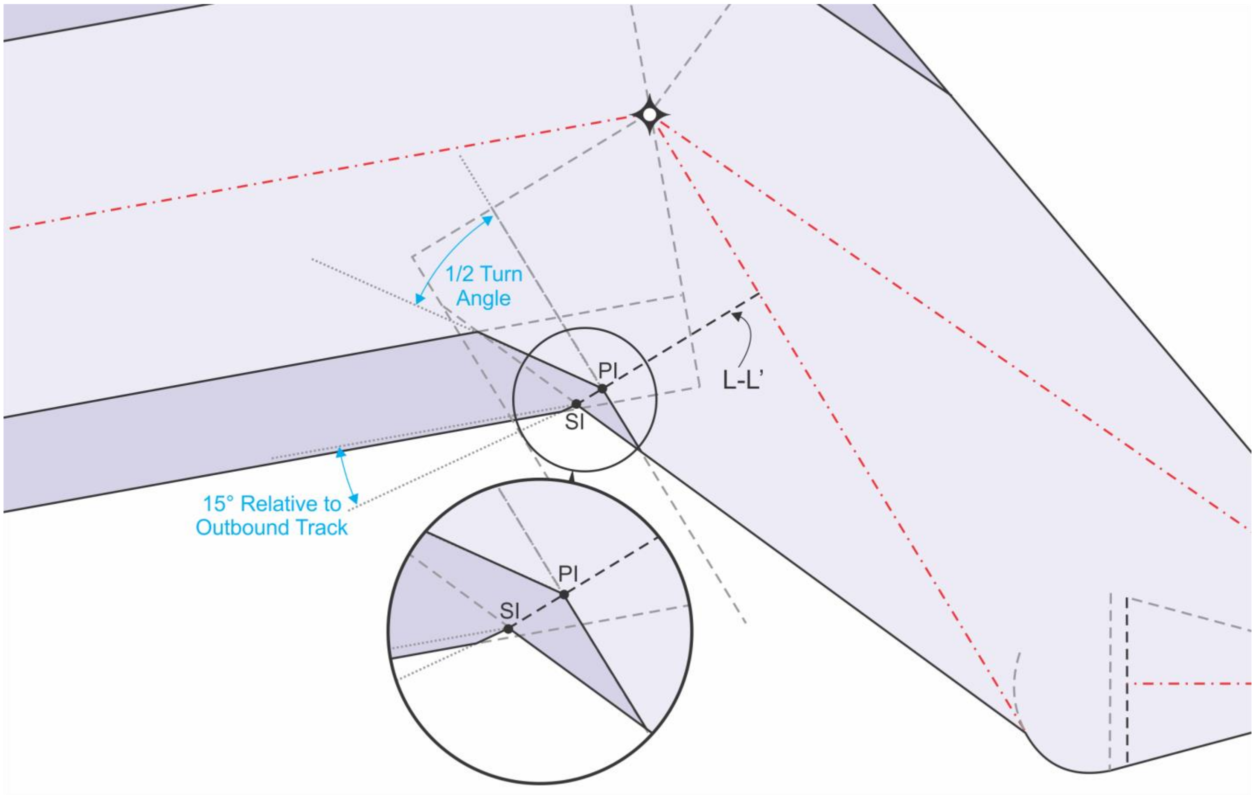
a Paragraph 5-3-1.a(1)(c)1 does not apply.

b For DF-TF, line L-L' is established perpendicular either on the early- or late-turn track, whichever is on the inside of the turn.

c When full primary and secondary width is reached at the early-turn baseline (L-L') construct in accordance with paragraph 1-2-5.

(b) Other connections. Construct in accordance with paragraph 1-2-5.

Figure 5-3-13. Early/Inside Turn, Second Turn, Turn at FB Fix DF-TF connection



b. Late and outside turn construction.

(1) Wind spiral application. Wind spiral (WS) construction applies to late and outside turns for turn-at-altitude, turn-at-fix (FO). See paragraph 1-3-1.g for design parameters.

(a) WS number and baselines. Baseline locations for WS construction may be dependent on Reaction and Roll distance [see formula 1-2-12 and table 5-3-4].

Table 5-3-4. Wind Spiral Number and distance to Late Turn Baseline

WS/Baseline Parameters	Turn-at-Fix (FO)	Turn-at-Altitude
WS Number	1 or 2	1, 2, or 3
WS1 & WS2 Distance Late-Turn Baseline (Line P-P') from turn fix/point	ATT + D_{rr}	ICA + D_{rr}
WS3 Baseline	ATT (early) for FO DF, otherwise not applicable	Parallel to DRL

1. First turn baseline. For first turn construction, the late-turn baseline (line P-P') marks the construction line for wind spirals. Line P-P' is located at the table 5-3-4 distance with points P and P' placed at the continuation of the inbound leg's outer boundary [see paragraph 1-3-1.g(4)]. The no-wind turn radius for the outside turn wind spiral (WS1) and inside turn wind spiral (WS2) will be measured perpendicular to the inbound track from their respective

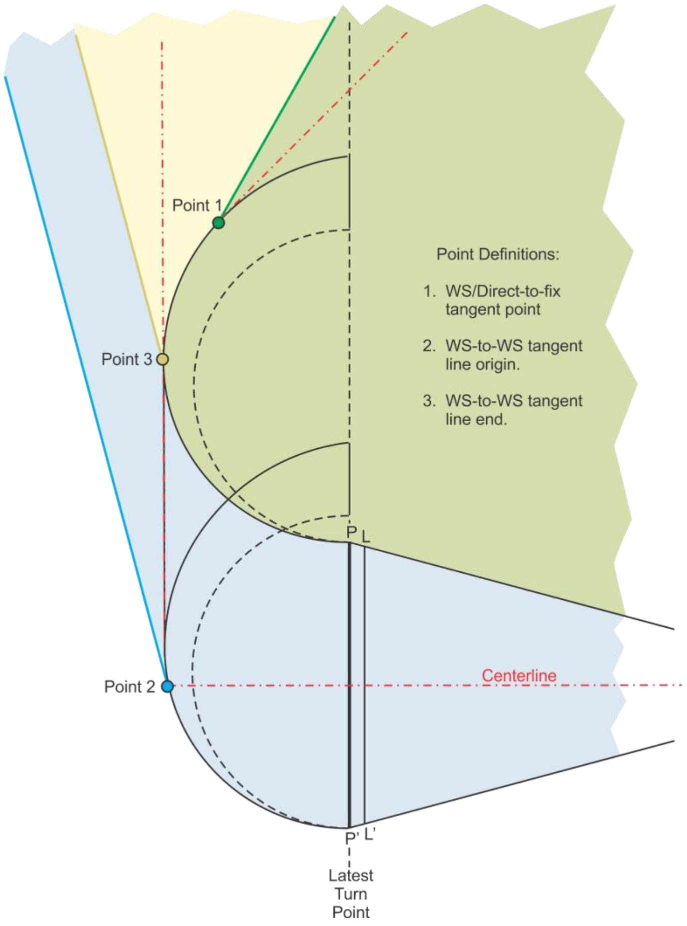
point P. If a third wind spiral is used (WS3), the baseline will be parallel to DRL with the no-wind turn radius measured inward from DRP, for DF-DF baseline will be line L-L'.

2. Second turn baseline [see figure 5-3-21]. To accommodate the two inbound tracks in the DF leg, the second turn construction uses two WS baselines, line P-P' for WS1 and line P'-P" for WS2. Each late turn baseline is oriented perpendicular to the early- and late-turn tracks at the table 5-3-4 distance. The baseline for the inbound track nearer the outside-turn boundary is designated line P'-P", with point P" placed on the extended outer boundary and point P' placed at the no-wind turn radius inward along the baseline from point P". The baseline for the inbound track nearer the inside-turn boundary is designated line P-P', with point P' placed on inbound track and point P placed on the extended outer boundary. The no-wind turn radius for the inside turn wind spiral is measured perpendicular to the inside track inward along the baseline from point P.

(b) WS connection point and outside turn OEA boundary. Each WS has various connection options along its path, which predicate the outside turn OEA boundary. The chosen connection must provide the most reasonably conservative track and protection area. Where excessive splay is required to reach full-width protection, consider lengthening the leg, restricting the speed, category, etc. to avoid protection and/or construction difficulties. Consider full-width protection to exist at the fix where the splay line is tangent to a full-width-radius circle about the fix.

1. Turn-at-altitude. For turns at an altitude, the 15-degree or greater splay line that joins the outbound leg outer boundaries may originate from the WS/direct-to-fix tangent point (point 1), the WS to WS tangent line origin (point 2) or the WS to WS tangent line end (point 3). Where the turn angle is ≤ 105 degrees, or the divergence angle between the WS to WS tangent line and the direct-to-fix line is ≤ 15 degrees, apply the splay line from the WS to WS tangent line origin. DF secondary areas begin/exist only where full width primary exists.

Figure 5-3-14. WS Connection Points, Turn-at-Altitude


Point 1
Point 3
Point 2
Point Definitions:
1. WS/Direct-to-fix tangent point
2. WS-to-WS tangent line origin.
3. WS-to-WS tangent line end.
P
L
Centerline
P'
L'
Latest Turn Point

2. Turn-at-fix (FO). For turns at a fix, the connection point and the OEA boundary is dependent on the WS boundary relative to the outbound leg OEA (extended). The connection point is the point of tangency on the WS in both cases. When the WS is not contained in the outbound leg OEA, the outside boundary is a 30-degree converging line relative to the outbound course. When the WS is contained in the outbound leg OEA, the outside boundary is a 15-degree or greater splaying line relative to the outbound course from the WS/outbound leg parallel point until reaching the outbound leg boundaries.

Figure 5-3-15. Late/Outside Turn, Wind Spiral, WS not contained in Outbound OEA

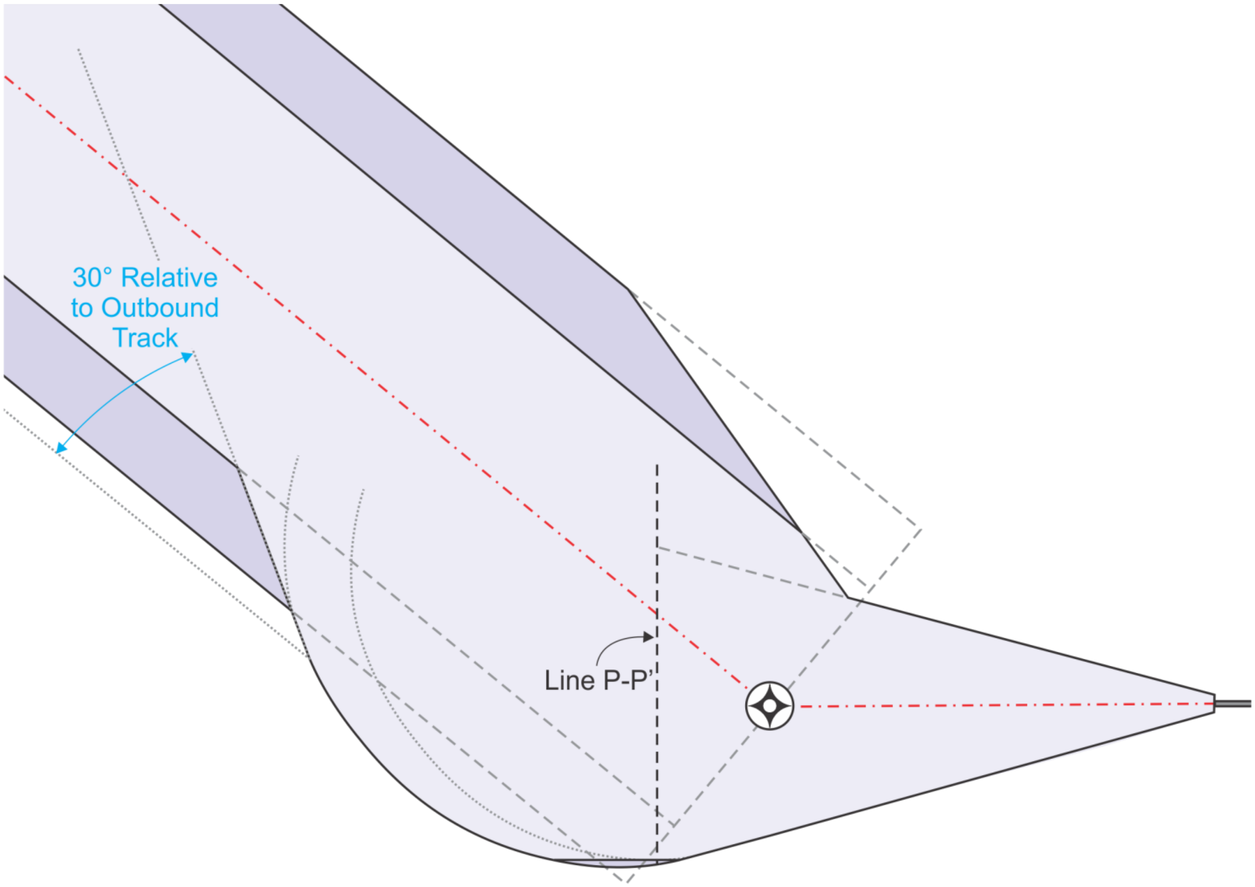
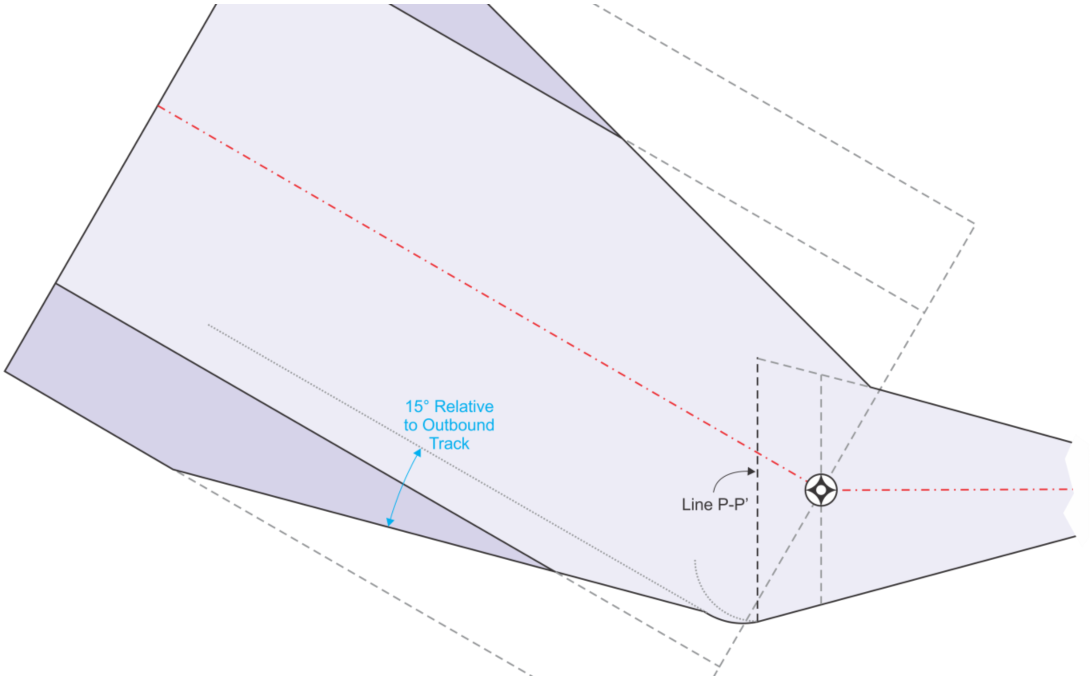


Figure 5-3-16. Late/Outside Turn, Wind Spiral, WS contained in Outbound OEA



(c) Determining multiple WS necessity. To determine multiple wind spiral necessity, construct the additional WS in the direction of turn from its prescribed location to its connection point. Where the additional WS intersects the preceding WS construction (including the connecting and expansion lines), connect the wind spirals with a tangent line that is parallel to the WS center points. Otherwise, revert to the previous WS construction.

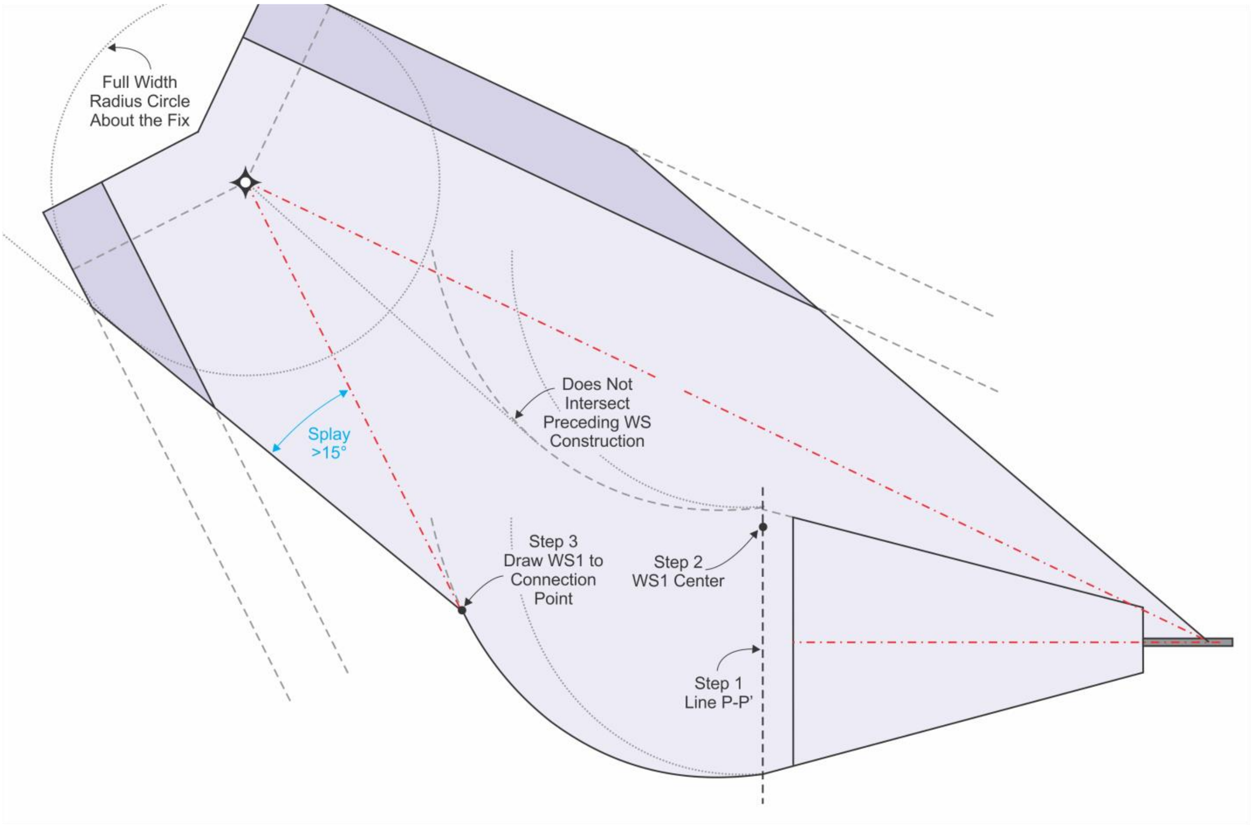
(2) First turn.

(a) Turn-at-altitude, turn-at-fix (FO), and minimum VI-CF construction.

1. *Step 1*. Construct the late-turn baseline (P-P') perpendicular to the runway centerline extended (or offset course if paragraph 5-2-1.a is used) at the late-turn-point. For minimum VI-CF construction late-turn point is 1 NM past the intercept point.

2. *Case 1*. Small turns using one WS.

Figure 5-3-17. Late/Outside Turn, First Turn, Case 1, Turn-at-Altitude or FO fix with 1 WS

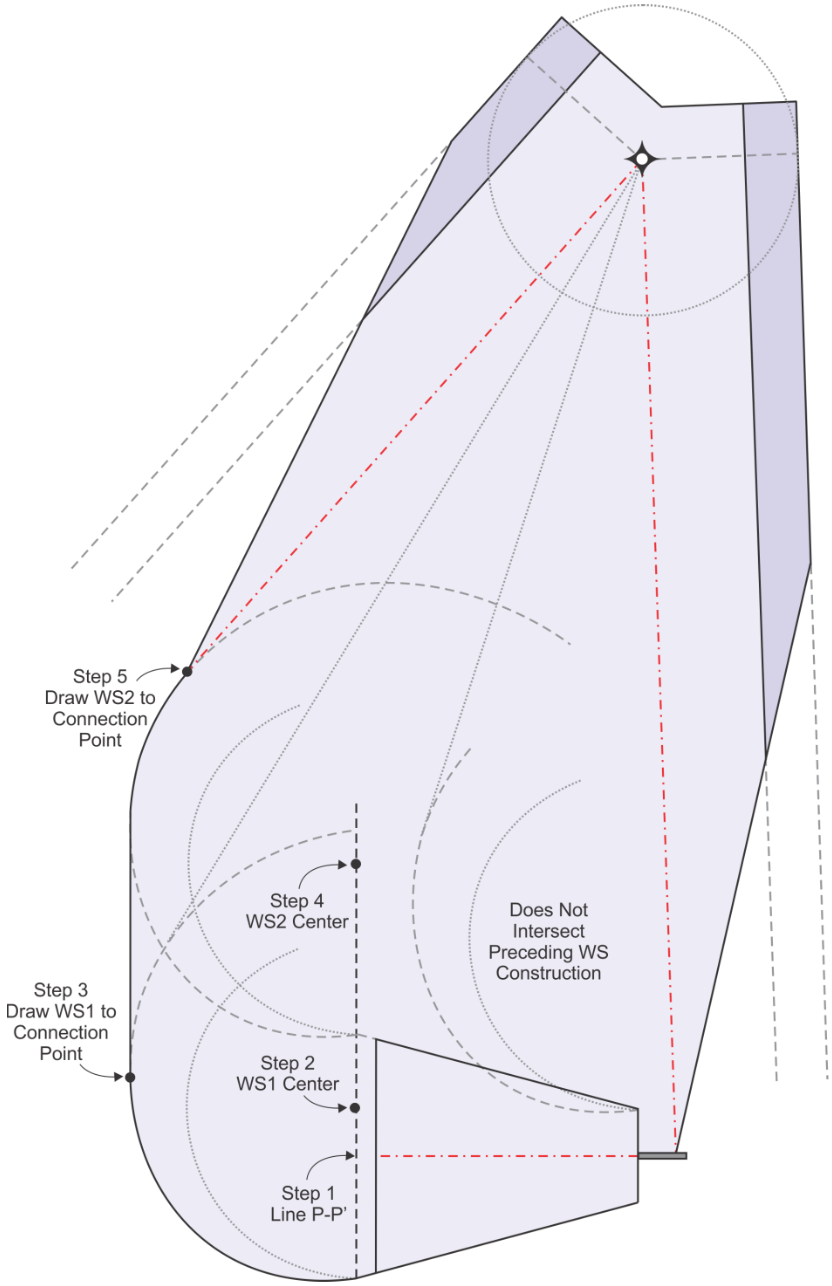


a *Step 2*. Locate the WS1 center on line P-P' at no-wind turn radius distance from point P'.

b *Step 3*. Construct WS1 from the outside turn point in the direction of turn until reaching its connection point. For minimum VI-CF construction use R (based on the highest projected turn altitude), place an arc center point at R distance from the outside turn point, construct an arc at distance R + 1 NM (this will result in a circle instead of a WS) [see figure 5-3-6].

3. *Case 2*. For turns nearing or greater than 90 degrees using more than one WS, complete *Steps 1-3* then:

Figure 5-3-18. Late/Outside Turn, First Turn, Case 2, Turn-at-Altitude or FO fix with 2 WS

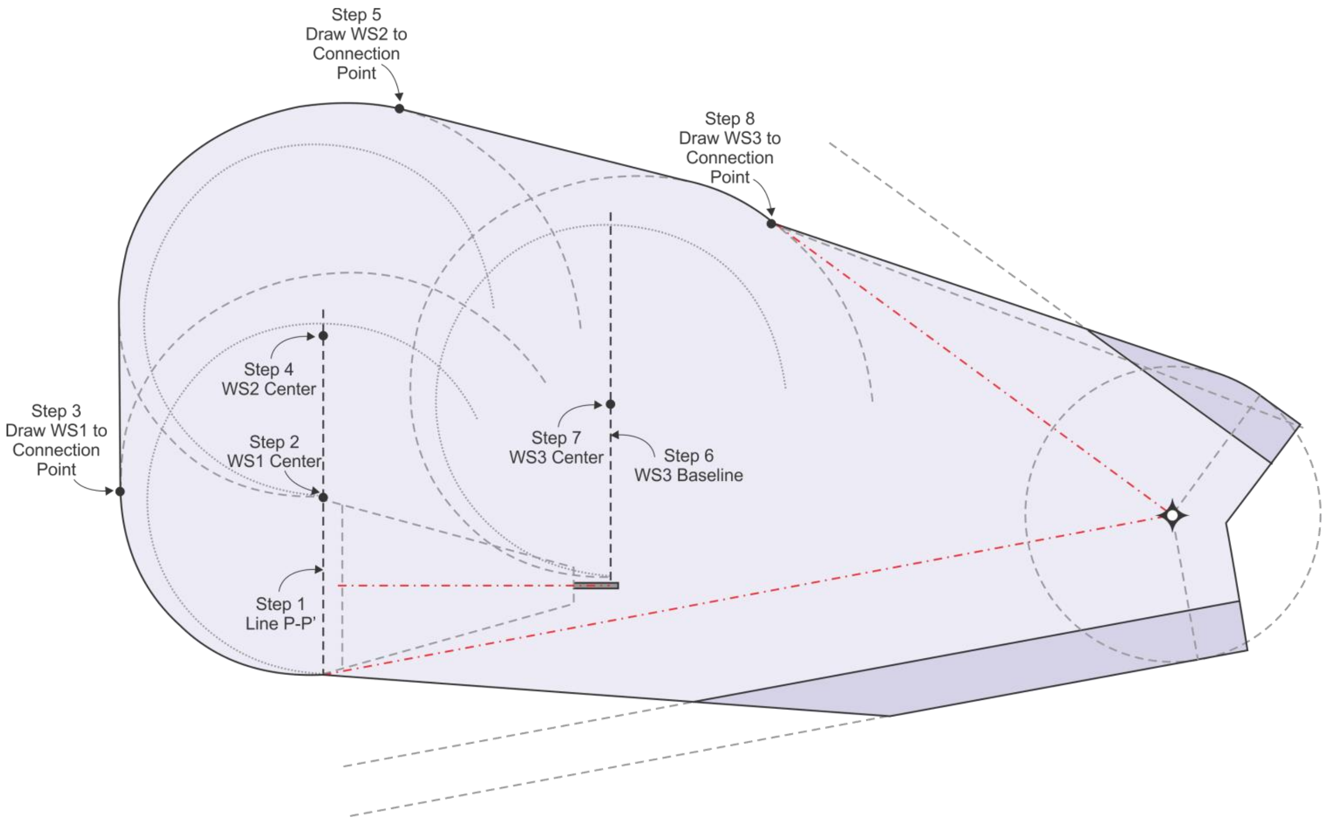


a *Step 4*. Locate the WS2 center on P-P’ at no-wind turn radius distance from point P.

b *Step 5*. Construct WS2 from the inside turn point in the direction of turn until reaching its connection point. For minimum VI-CF construction use R (based on the highest projected turn altitude), place an arc center point at 2 × R distance from the outside turn point, construct an arc at distance R + 1 NM (this will result in a circle instead of a WS) [see figure 5-3-7].

4. *Case 3*. Turns nearing or greater than 180 degrees using more than two WS, complete *Steps 1-5* then:

Figure 5-3-19. Late/Outside Turn, First Turn, Case 3, Turn-at-Altitude with 3 WS



a *Step 6*. Construct the WS3 baseline perpendicular to the runway centerline along DRL extended toward the turn side. For FO DF the WS3 baseline is on line L-L' perpendicular to the runway centerline extended toward turn side.

b *Step 7*. Locate the WS3 center on the baseline at no-wind turn radius distance from DRP.

c *Step 8*. Construct WS3 from this point in the direction of turn until reaching its connection point.

(b) Turn-at-fix (FB) and standard VI-CF construction.

1. *Step 1*. Construct the outer primary boundary using a radius of 2 × segment XTT centered on the plotted fix position, truncated at the inbound leg extended primary boundary until tangent to the outbound leg primary boundary (not applicable to VI-CF).

2. *Step 2*. Construct the secondary boundary using a radius of 3 × segment XTT entered on the plotted fix position, truncated at the inbound leg extended outer boundary until tangent to the outbound leg outer boundary. For VI-CF this will be the outer primary boundary connecting to point X then parallel to CF course to point W [see figure 5-3-3].

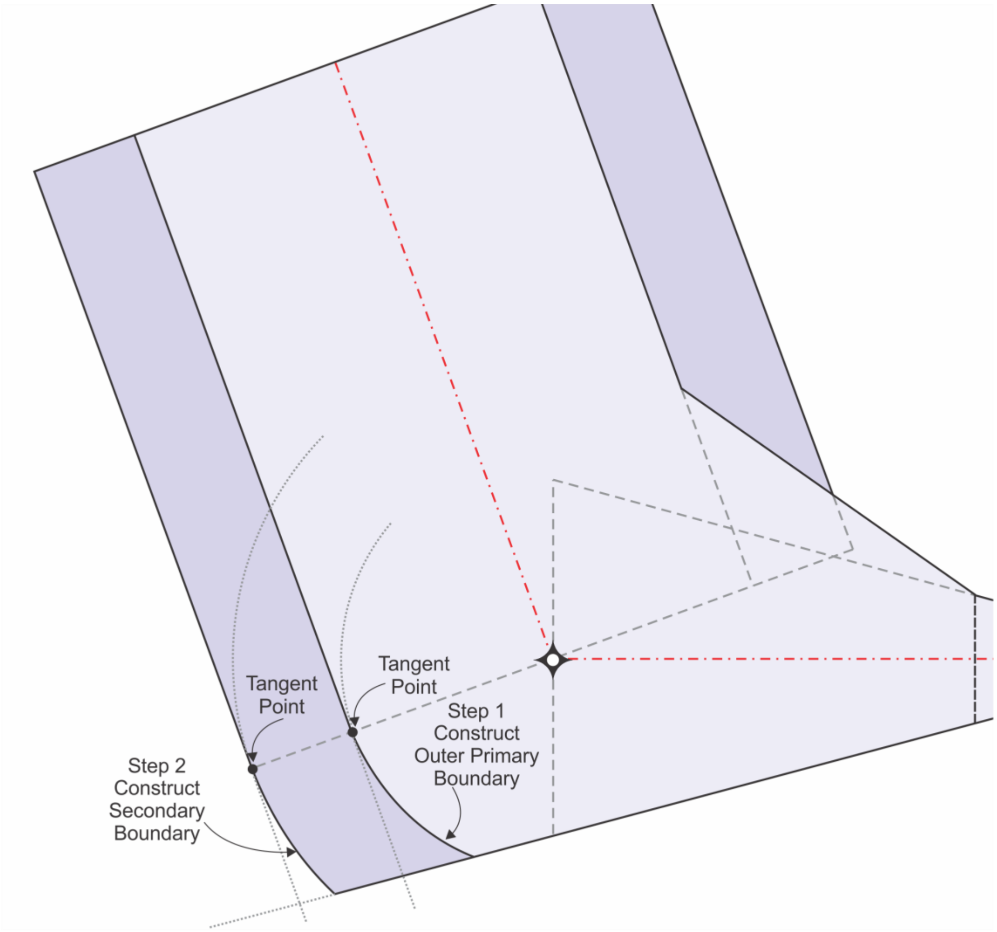
(c) Turn-at-fix, RF construction.

1. *Step 1.* Construct the outer primary boundary by truncating the RF full width primary area at the initial straight leg extended splay line [see figure 5-3-12].

2. *Step 2.* Construct the secondary boundary by truncating the RF full width secondary area at the initial straight leg extended splay line [see figure 5-3-12]

3. Where the initial straight leg splay hasn't reached a width of 3 × segment XTT by the end of the first RF and the succeeding legs is a TF or RF with same direction turn, the initial straight leg splay may continue into the next leg until full width is reached [see figure 5-3-11]; otherwise, increase splay as necessary to reach full OEA width by the end point of the first RF.

Figure 5-3-20. Late/Outside Turn, First Turn, FB fix


Tangent Point
Tangent Point
Step 1 Construct Outer Primary Boundary
Step 2 Construct Secondary Boundary

(3) Second turn.

(a) Turn-at-fix (FO).

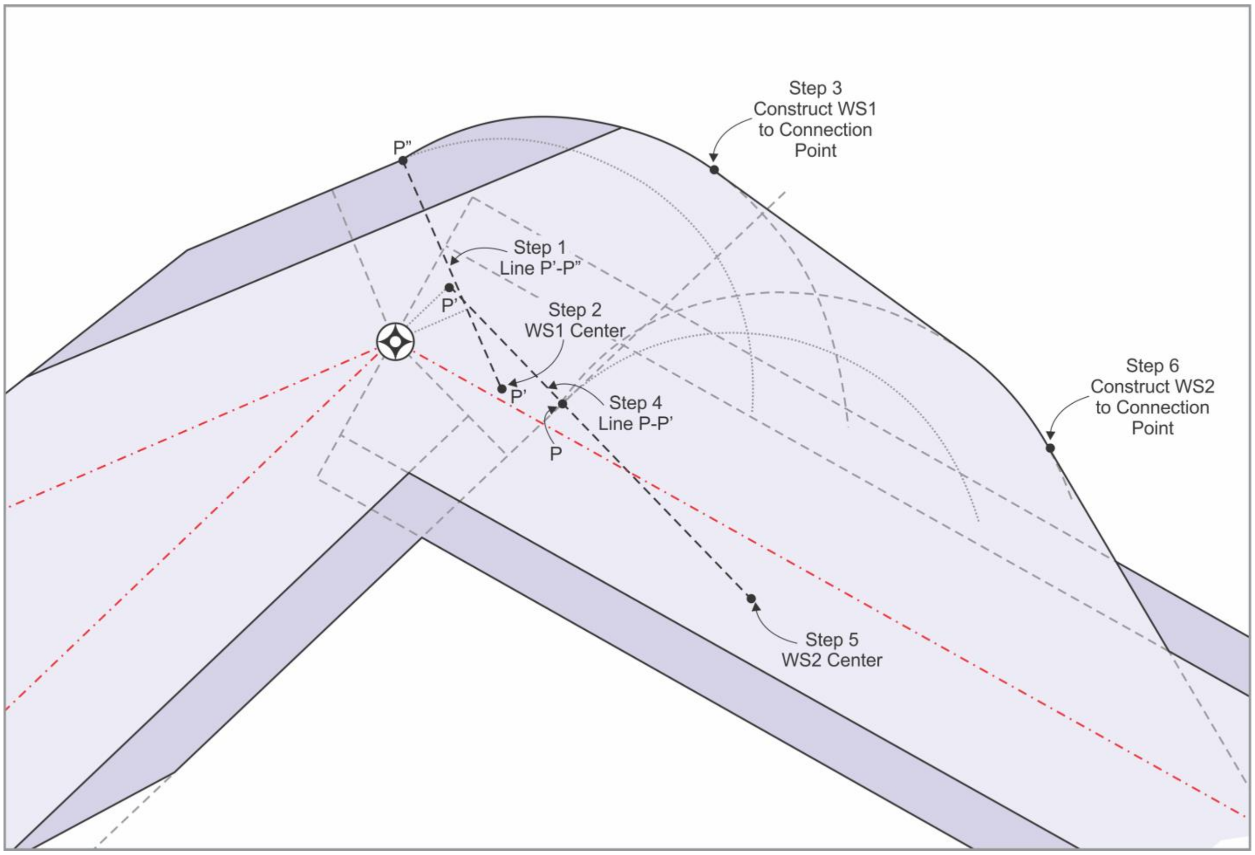
1. *Step 1*. Construct the WS1 baseline, (line P'-P") perpendicular to the DF track nearer the outside of the DF-TF turn at the late turn point.

2. *Step 2*. Locate the WS1 center on line P'-P" at no-wind turn radius distance from point P."

3. *Step 3*. Construct WS1 the outside point in the direction of turn until reaching its connection point.

4. *Step 4*. Construct the WS2 baseline, (line P-P') perpendicular to the DF track nearer the inside the DF-TF at the late turn point.

Figure 5-3-21. Late/Outside Turn, Second Turn, FO fix



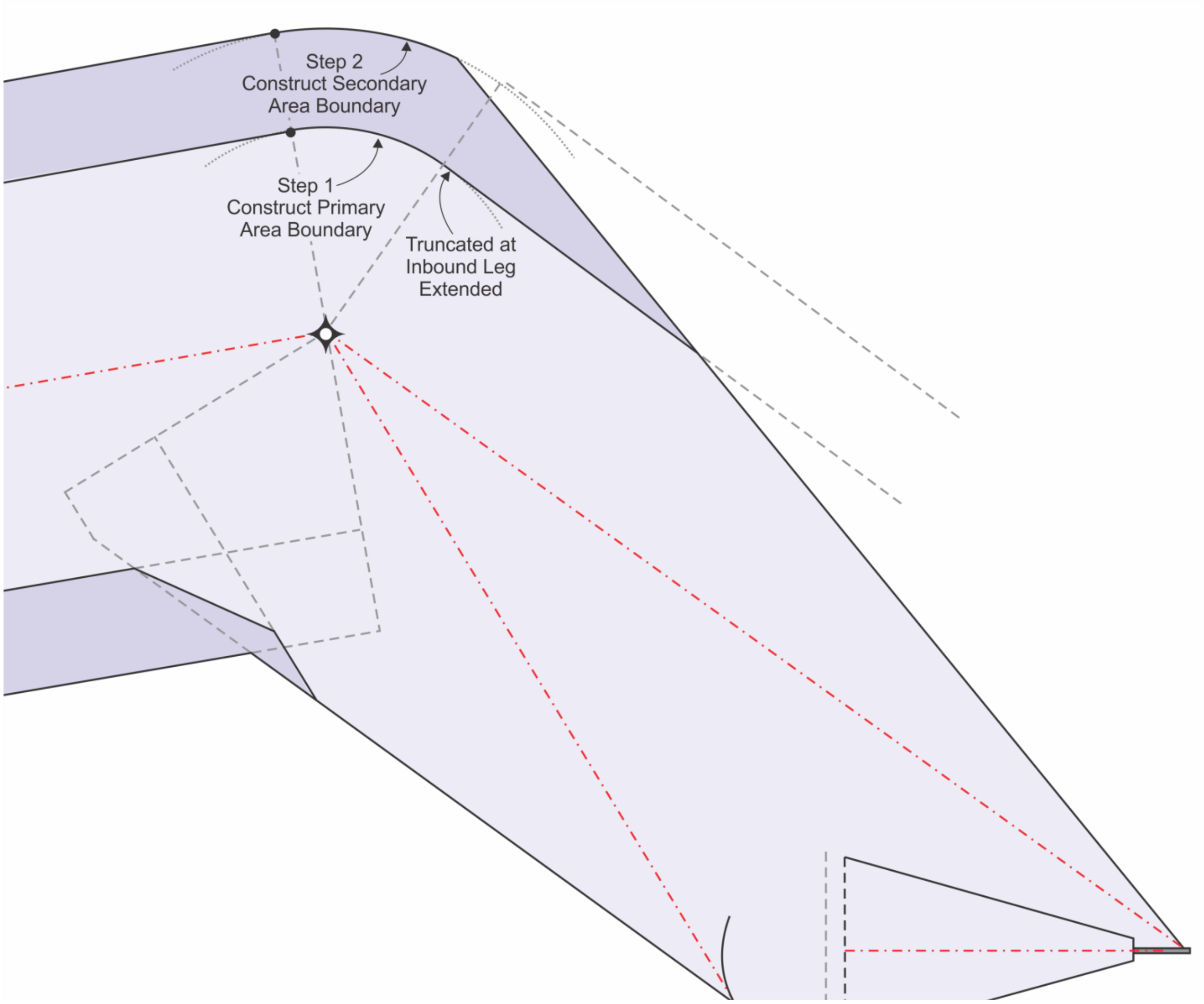
5. *Step 5*. Locate the WS2 center on line P-P' at no-wind turn radius distance from point P.

6. *Step 6*. Construct WS2 from the inside turn point in the direction of turn until reaching its connection point.

(b) Turn-at-Fix (FB).

1. When the inbound leg outside boundary is less than full primary and secondary width reached perpendicular to the fix, construct in accordance with paragraph 5-3-1.b(2)(b) [see figure 5-3-16].

Figure 5-3-22. Late/Outside Turn, Second Turn, Turn at FB fix, less than full OEA width perpendicular to fix



2. When the inbound leg outside boundary is full primary and secondary width perpendicular to the fix, construct in accordance with paragraph 1-2-5.

Section 5-4. Obstacle Evaluation

5-4-1. Obstacle evaluations.

a. Utilize the general concepts from paragraph 1-3-1.g. Where an obstacle requires multiple measurements (an obstacle is equidistant from multiple primary boundary points, it lies along perpendiculars from multiple primary boundary points, etc.), apply the most adverse result from each of the combined primary/secondary measurements [see figure 5-4-1 through figure 5-4-7].

b. Primary OCS.

(1) ICA. Measure distance to obstacles using the distance from ICAB along runway centerline extended. The ICA OCS begins at the MSL elevation of the DER [see formula 1-3-5 to determine OCS slope].

(2) Straight departure. Measure and apply the OCS slope from ICAB along the track distance to a point at/abeam the obstacle.

(3) Turn-at-altitude (single and multiple legs). Apply the OCS slope along the shortest primary area distance from the ICA boundary to a point at/abeam the obstacle.

(a) Section 1w. For obstacles past the DER, measure from the closest point on the ICA boundary. For obstacles between the DRL and DER, take the lesser of the distance from runway centerline and the closest point on the ICA boundary. The section 1 OCS begins at the MSL elevation of the OCS at ICAE.

(b) Section 2. Measure from DRP. The section 2 OCS begins at airport elevation plus 500 (for LNAV engagement).

(4) Turn-at-a-fix (single and multiple legs). Apply the OCS slope from ICAB (parallel to track) to L-L’, then along the shortest primary distance to a point at/abeam the obstacle.

(5) VI-CF.

(a) VI-CF standard construction and minimum VI-CF with turns of 30 degrees or less. Apply the OCS slope along the shortest primary area distance from the ICA boundary to a point at/abeam the obstacle.

(b) Minimum VI-CF with turns of more than 30 degrees. For obstacles past the DER, measure from the closest point on the ICA boundary. For obstacles between the DRP and DER, take the lesser of the distance from runway centerline and the closest point on the ICA boundary.

(6) Routes. Apply ICA, section 1 and section 2 methodology where applicable, then measure along the shortest primary area distance that passes through the early-turn baseline of all preceding legs.

Note: The shortest primary area distance is the length of the shortest line kept within primary area that passes through the early-turn baseline of all preceding legs.

c. Secondary OCS. For obstacles located in secondary areas, calculate the primary OCS elevation as stated, then apply a 12:1 OCS slope along the shortest secondary distance to the obstacle. In straight legs, this is normally perpendicular to the nominal track. In expansion areas, the slope rises in a direction perpendicular to the primary boundary (arc, diagonal corner-cutter, etc.).

Figure 5-4-1. VA-DF Obstacle Distance Measurements, Primary and Secondary Area, large turn

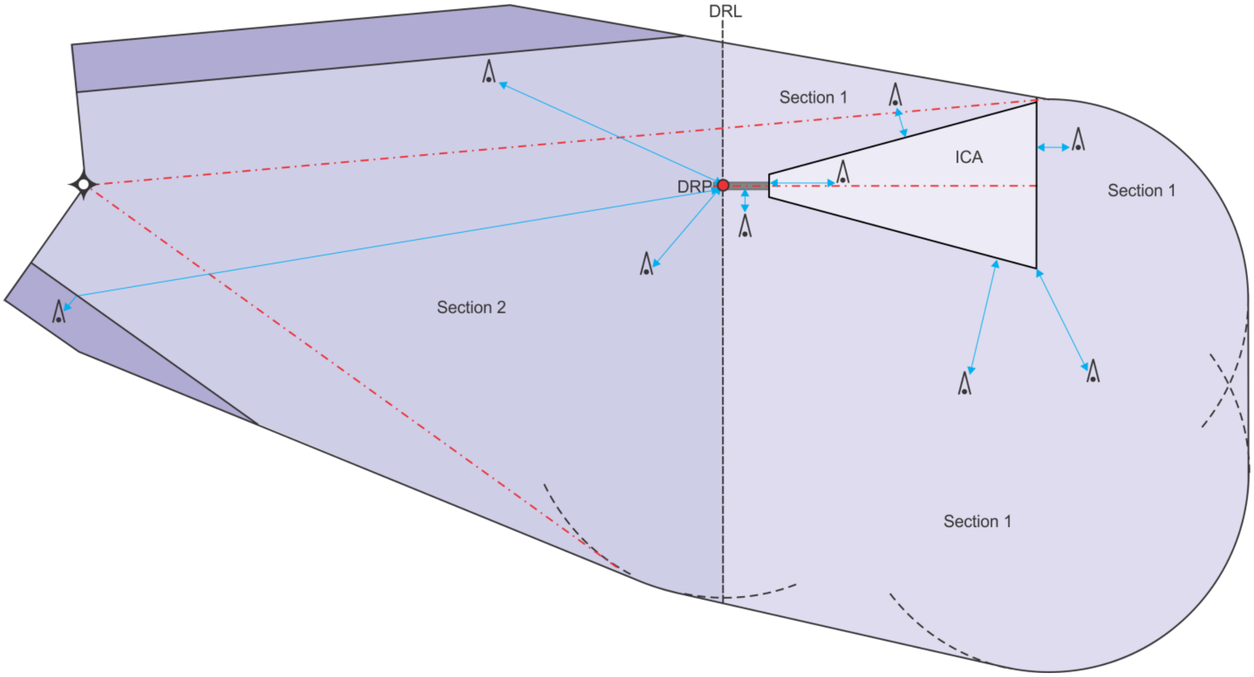


Figure 5-4-2. Obstacle Distance Measurements, Turn at Altitude, small turn

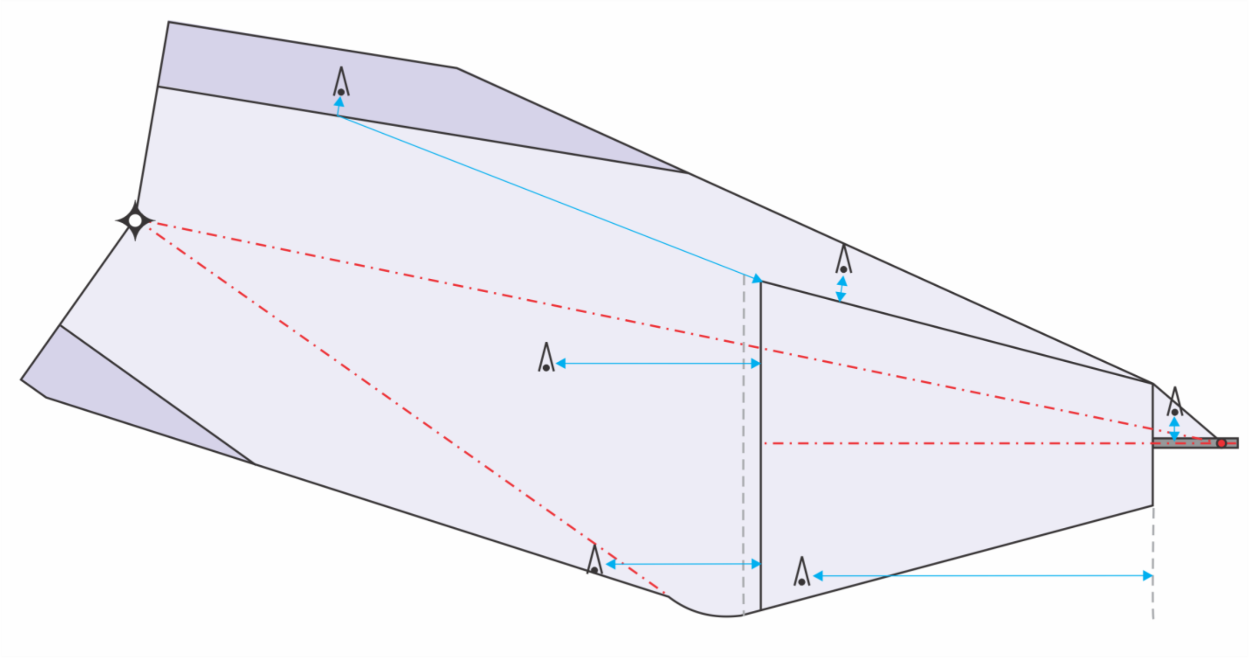


Figure 5-4-3. Obstacle Distance Measurements, Turn at Altitude, multiple legs

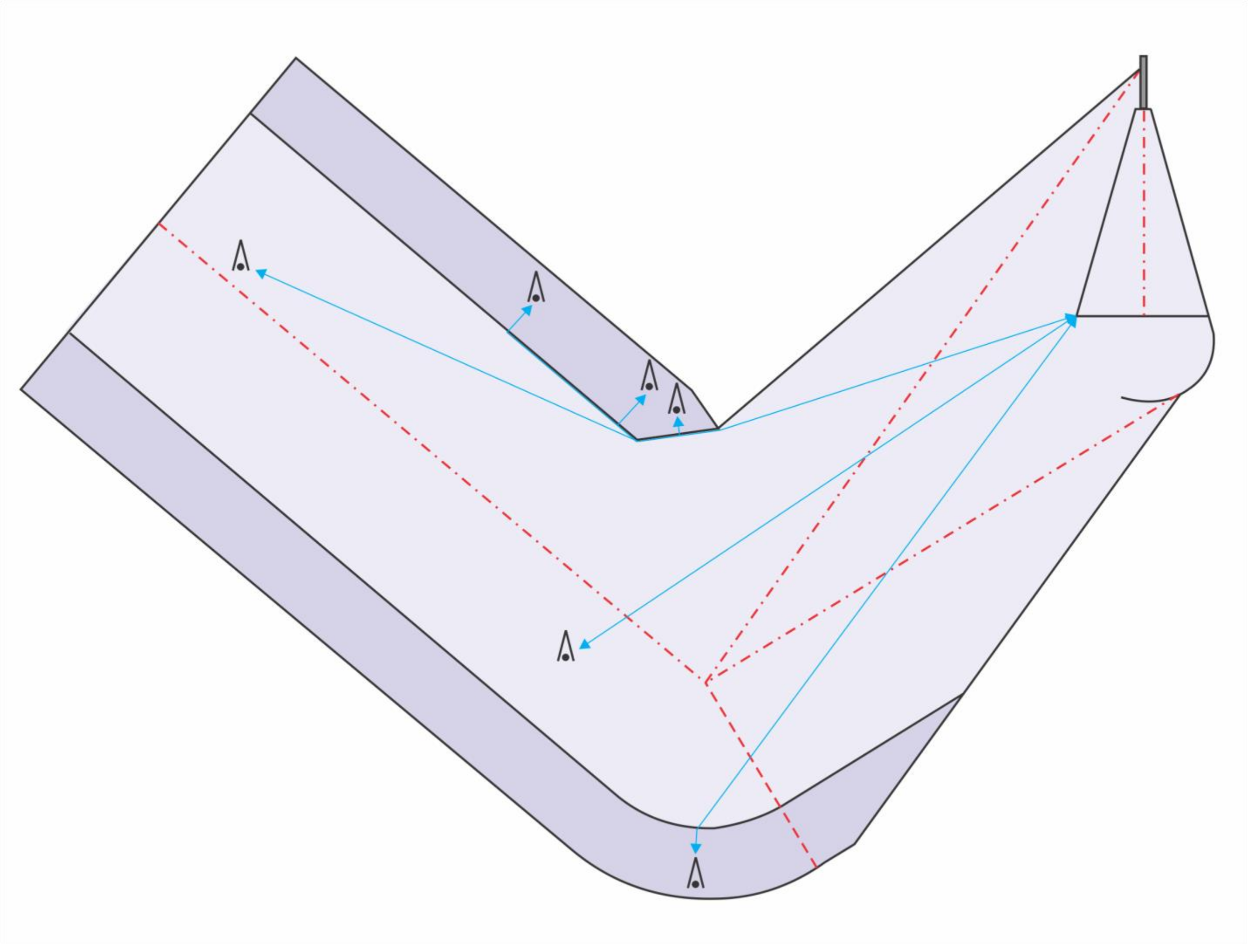


Figure 5-4-4. Obstacle Distance Measurements, Turn at FB Fix

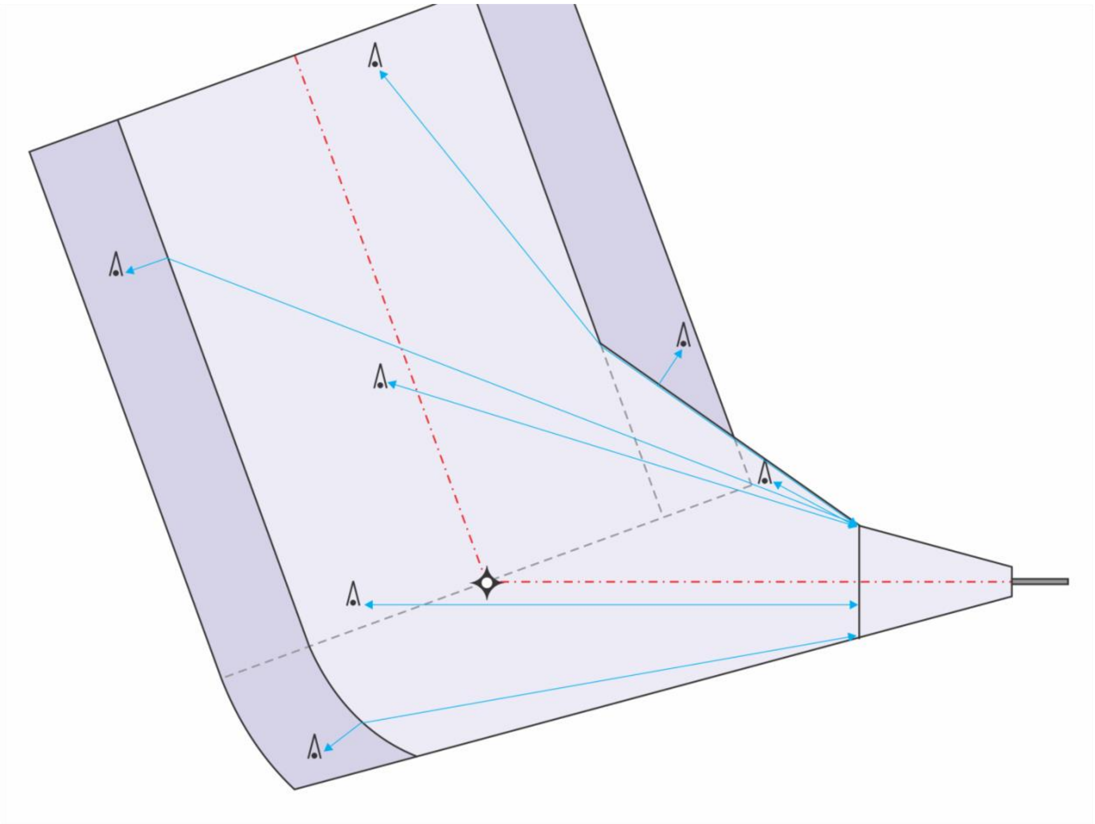


Figure 5-4-5. DF-DF Obstacle Distance Measurements, Turn at FO Fix

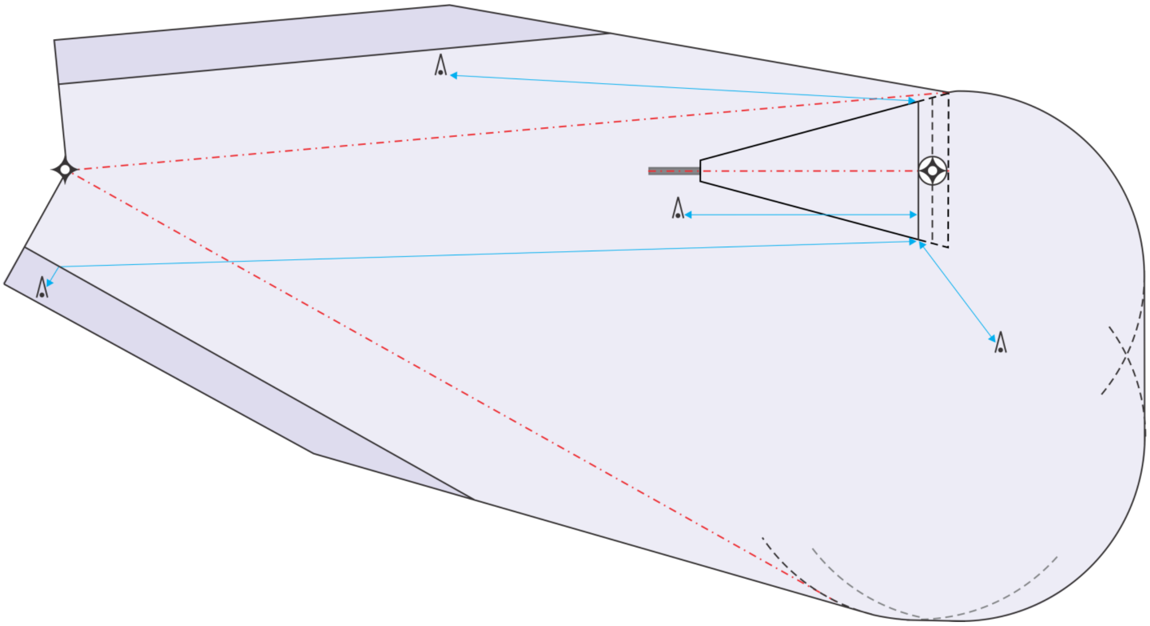


Figure 5-4-6. VI-CF Obstacle Distance Measurements

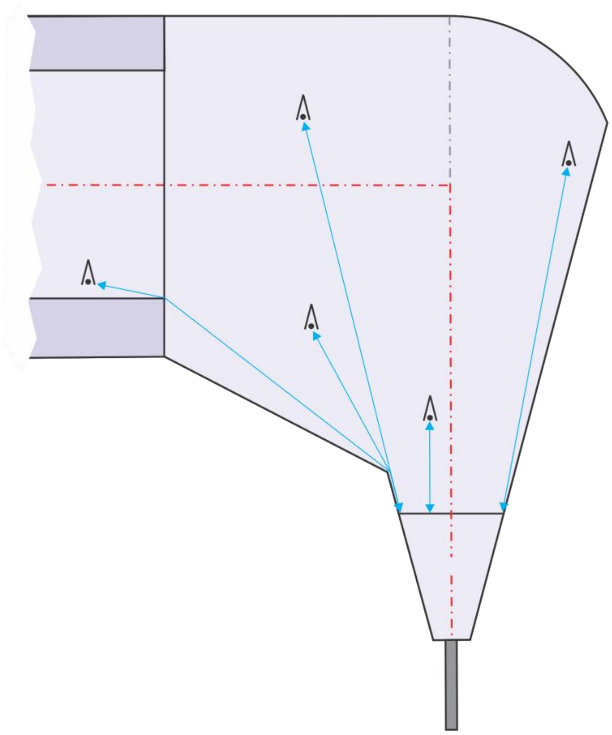
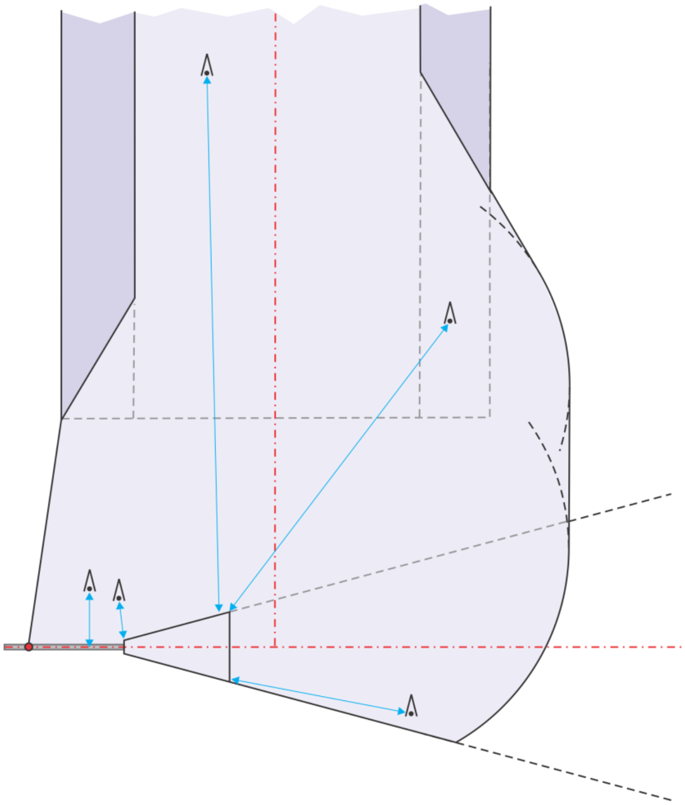


Figure 5-4-7. Minimum VI-CF with turn greater than 30 degrees Obstacle Distance Measurements



Section 5-5. Climb Gradient

5-5-1. Climb gradient. Where the default OCS slope is penetrated, a greater than standard CG may be required to clear the penetrating obstruction. Multiple higher than standard CGs and/or a CG greater than 500 ft/NM requires Flight Standards approval.

a. When climb gradients are used for obstacle clearance, compute the climb gradient per paragraph 1-3-1.g and formula 1-3-7 using the CG termination altitude from formula 5-5-1.

Formula 5-5-1. CG Termination Altitude

$$\mathrm{CG_{term}} = \frac{\mathrm{O_{MSL}} - \frac{\mathrm{d_{primary}}}{12} - \mathrm{DER_{elev}}}{0.76} + \mathrm{DER_{elev}}$$

Where:
$\mathrm{DER_{elev}}$ = departure end of runway elevation (feet)
$\mathrm{d_{primary}}$ = dist. (feet) from edge of primary to obstacle (zero if not in secondary)

Example:

$$\mathrm{CG_{term}} = \frac{2147.41 - \frac{390.12}{12} - 1104.3}{0.76} + 1104.3$$

$\mathrm{CG_{term}} \approx 2434.04$ feet

b. Mitigating obstacle penetrations. The preferred method of obstacle mitigation is to use a less onerous route. The next choice is a climb gradient, using paragraph 1-3-1.g(2). Also see Order 8260.3, chapter 14 and Order 8260.46.

c. Climb in a holding pattern. Where required, apply climb-in-hold criteria contained in Order 8260.3, section 17-7.

Section 5-6. Radar Vector SID

5-6-1. General. When operationally advantageous, a SID may be developed starting with or including a vector leg followed by an RNAV route. [See appendix D for implementation considerations.]

5-6-2. SID starting with vector leg. When designing an RNAV route following radar vectors, establish the first waypoint as an Initial Fix leg (IF). No OEA construction or DME/DME screening is required prior to the IF of the RNAV route being joined.

a. The IF beginning an RNAV route must fall within an area and be at an altitude protected by an MVA/MIA. The IF altitude may not be less than 500 feet above airport elevation. ATC must be consulted and consideration must be given to the time/distance required for RADAR identification and normal vectoring when designing the IF.

b. A single RNAV course must be defined from the IF.

c. The IF must be a FB fix. A TF leg must follow the IF. General OEA construction applies [see paragraph 1-2-5].

d. The length of the first leg must be sufficient to accommodate a 90-degree turn at the IF. Use standard turn parameters. Where a shorter leg is needed, reduce airspeed in increments of not less than five KIAS until the desired length is achieved [see table 1-2-2]. An alternative method to allow design of a shorter first leg is to limit the amount of intercept to 45 degrees, this must be approved and applied by ATC.

e. Each SID is limited to one common route and must start with a single IF.

f. OCS slope and origin. A flat surface evaluation is conducted from the IF early ATT to late ATT. A sloping OCS [see formula 1-3-5] originates at IF late ATT. The OCS starting elevation is the MVA/MIA altitude minus ROC. Where multiple MVA/MIA sectors apply, the most demanding surface must be used. Climb gradients are not authorized as mitigation for obstacles that penetrate the flat or sloping OCS, the surface must be raised by increasing fix altitudes or redesigning the route.

5-6-3. SID including vector legs. A SID may start with an RNAV route that has a manual termination followed by vectors to join an RNAV route (Open SID). Construct and evaluate the route off the runway using the applicable straight or turning criteria. No OEA construction or DME/DME screening is required between the fix beginning a manual termination leg and the IF of the RNAV route being joined.

a. The RNAV route off the runway must terminate with either an FM or VM.

b. An FM beginning with a FB fix is preferred. This combination encompasses the most operators.

c. An FM that results in a turn must start with a FB fix.

d. An FM that doesn’t require a turn may be either a FB or FO fix.

e. A VM must start with a FO.

f. The fix beginning a manual termination leg must fall within an area and at an altitude protected by either an MVA/MIA or DVA.

(1) When a sectored DVA is used, the heading/course of the manual termination leg must comply with the heading limitations of that DVA sector.

(2) If the manual termination leg starts with a FO fix and a sectored DVA is used, the turn to the heading/course must be contained within that DVA sector.

g. For an RNAV route following a vector segment, comply with paragraphs 5-6-2.a thru 5-6-2.f.

THIS PAGE IS INTENTIONALLY LEFT BLANK

Appendix A. Use of Historical Wind Studies for Determining Tailwind

1. Purpose. This appendix addresses the use of five-year historical wind analysis to determine historical tailwind in lieu of calculated tailwind.

2. Acceptable source. Historical winds must be obtained or derived from the NOAA/NCEP hourly operational weather prediction system numerical forecast models (i.e., Rapid Update Cycle (RUC) or Rapid Refresh (RAP).

a. Use wind data from the last five-years of "zero-hour" data (best estimate of current conditions taken from aircraft measurements, weather stations, radar winds, and satellite imagery system).

b. Project the wind data on a Lambertian Conformal Hybrid-Isentropic grid at a resolution of not more than 20 km.

c. Derive the 99th percentile wind speeds by altitude using linear interpolation from the nearest grid point(s) to the applicable turn waypoint. For FB or FO turns, use the fix location for wind. For RF use the turn center location for wind.

d. 99th percentile winds speeds may be further sub-divided by direction.

3. Use of directional wind speeds. Historical winds sub-divided by direction may be used to support TF/FB, TF/FO, and RF turn construction only. Use the greatest 99th percentile wind speed from any direction that results in a tailwind during the turn (i.e., the "tailwind direction sector").

a. For TF legs, the turn start and end point is the FB or FO waypoint. For RF legs, the turn start point is the RF leg start waypoint and turn end point is the RF leg end waypoint.

b. For TF/FB, TF/FO, and RF legs, the start bearing is the reciprocal of inbound leg at the turn start point.

c. For TF/FB and RF turns, the end bearing is the reciprocal of the outbound leg at turn end point.

d. For TF/FO turns, the end bearing 30 degrees from the reciprocal of the outbound leg at turn end point.

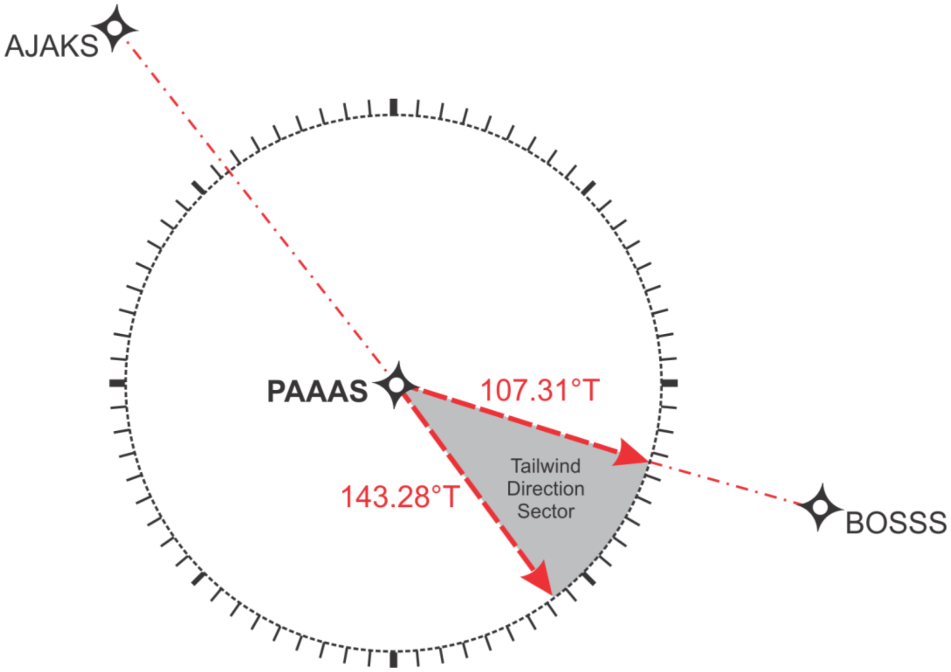
e. Examples:

(1) TF/FB turn: KDEN ILS or LOC Rwy 34L, tailwind direction sector for turn at PAAAS [see figure A-1]

(a) Start bearing; reciprocal of inbound leg at PAAAS = 107.31 (true).

(b) End bearing; reciprocal of outbound leg at PAAAS= 143.28 (true).

Figure A-1. TF/FB Turn Tailwind Direction Sector

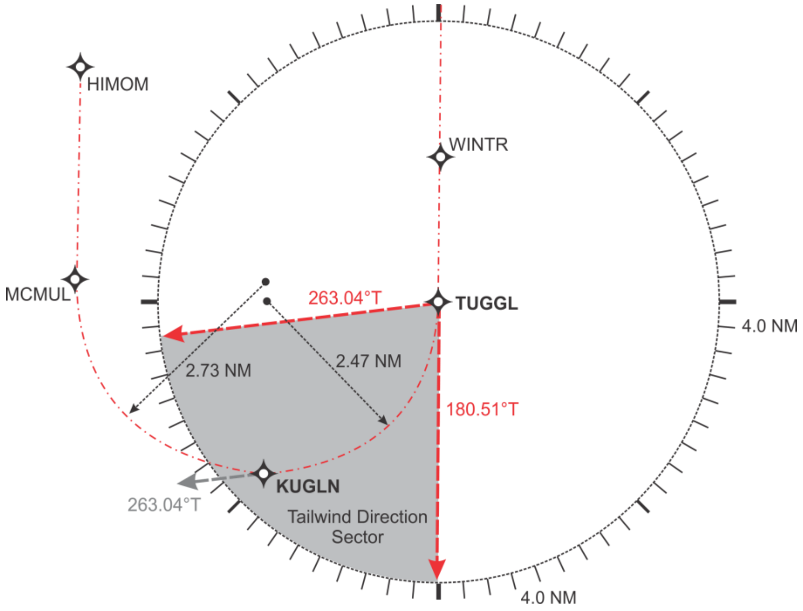


(2) RF leg: KDEN RNAV (RNP) Z RWY 34L, tailwind direction sector for KUGLN – TUGGL [see figure A-2].

(a) Start bearing; reciprocal of inbound leg at KUGLN = 263.04 (true).

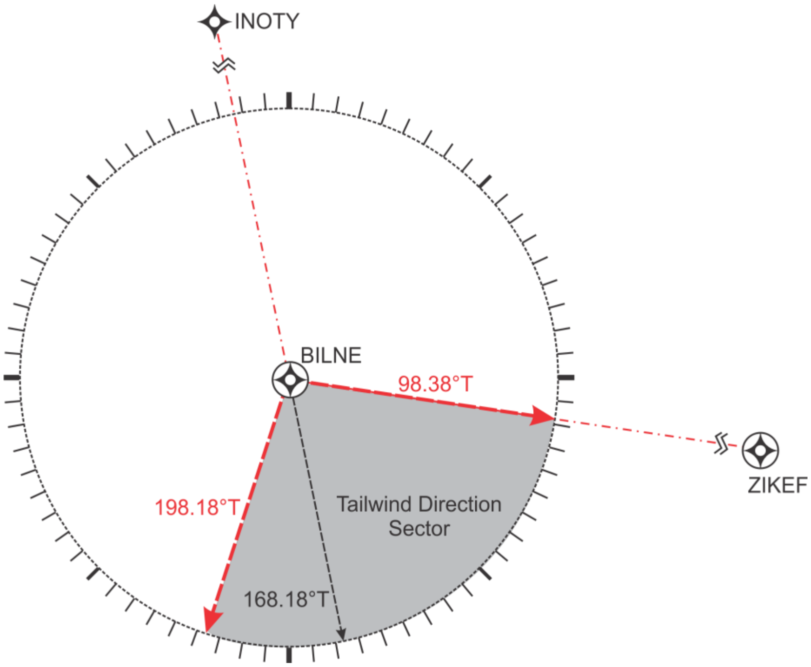
(b) End bearing; reciprocal of outbound leg at TUGGL= 180.51 (true).

Figure A-2. RF Leg Tailwind Direction Sector



(3) TF/FO turn: PAAK INOTY ONE (RNAV) Departure, tailwind direction sector for turn at BILNE [see figure A-3].

Figure A-3. TF/FO Turn Tailwind Direction Sector



THIS PAGE IS INTENTIONALLY LEFT BLANK

Appendix B. DF Leg Feasibility Analysis

1. Purpose. This analysis checks the DF leg for feasibility and is separate from the OEA evaluation. It is only intended to be a preliminary screening for flyability and does not ensure OEA construction.

2. Applicability. This analysis applies to DF legs used in departures and missed approach.

a. DF following turn -at-altitude.

(1) Determine the earliest turn point (TP).

(a) Departure. Measure from the DRP at airport elevation with 1100 ft/NM climb gradient until reaching the earliest climb-to altitude or the DER, whichever occurs first. If the climb-to altitude is not reached by the DER, continue the climb determination starting at the DER using a climb gradient at 500 ft/NM until 10000 feet, then 350 ft/NM until 18000 feet, then 200 ft/NM above 18000 feet.

(b) Missed approach. The earliest TP is line C-D.

(2) Determine the latest TP.

(a) Departure. Commencing at the DER at DER elevation, the latest TP is where an aircraft reaches the climb-to altitude at a climb gradient of 200 ft/NM or the minimum climb gradient required for obstacle clearance whichever is higher.

(b) Missed approach. Commencing at the aircraft [see paragraph 3-7-1.b], the latest TP is where an aircraft reaches the climb-to altitude at a climb gradient of 200 ft/NM or the minimum climb gradient required for obstacle clearance whichever is higher.

(3) Determine the turn radius. Use the highest altitude in the turn [see paragraph 1-2-5.c].

b. DF following FO fix (applicable to departure only).

(1) Determine the turn points. The early TP is ATT prior to the FO fix. The late TP is ATT + reaction and roll distance [see formula 1-2-12] after the FO fix.

(2) Determine the turn radius. Use the assumed altitude at the fix [see paragraph 1-2-5.c].

3. Analysis. Given the location of the DF leg termination fix and the outbound track from this fix, verify each of the following. When all three conditions are met, the DF leg is feasible.

a. DF fix location. The fix is on or outside all paths from the earliest TP until the latest TP based on the specified turn radius.

b. Turn following fix. The turn at the fix is 90 degrees or less.

c. DTA. Where the fix is a FB, the required DTA is available.

Figure B-1. DF Following Turn-at-altitude

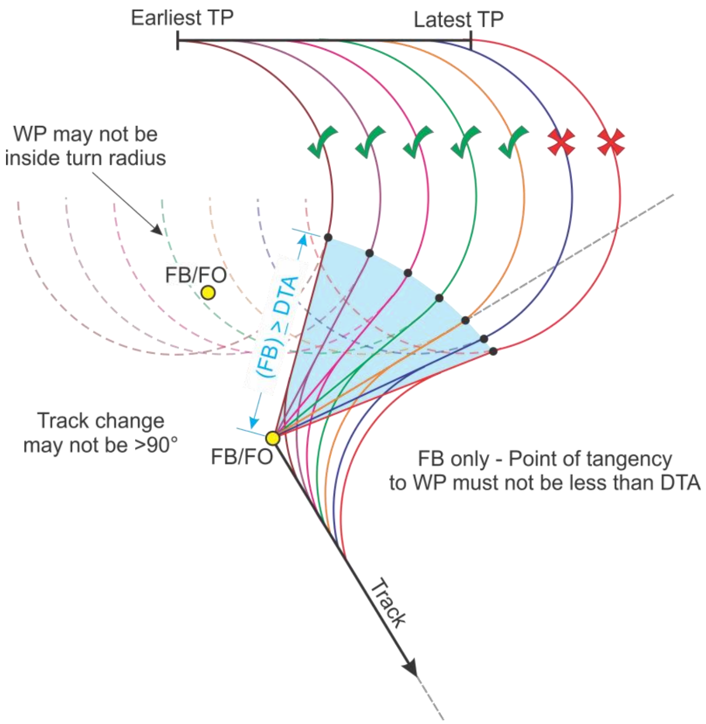
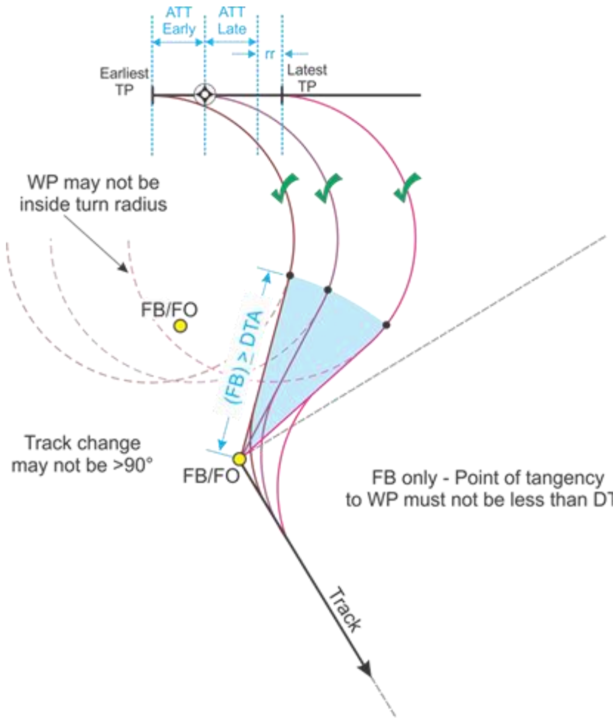

Earliest TP
Latest TP
✓ ✓ ✓ ✓ ✓ ✗ ✗
WP may not be inside turn radius
FB/FO
(FB) ≥ DTA
Track change may not be >90°
FB/FO
FB only - Point of tangency to WP must not be less than DTA
Track

Figure B-2. DF Following FO fix


ATT Early
ATT Late
rr
Earliest TP
Latest TP
✓ ✓ ✓
WP may not be inside turn radius
FB/FO
(FB) ≥ DTA
Track change may not be >90°
FB/FO
FB only - Point of tangency to WP must not be less than DTA
Track

Appendix C. PBN Transition to ILS/GLS Final

1. Purpose. When establishing an RF intermediate segment that transitions to an ILS/GLS final, the design must account for high temperature conditions that may cause higher than indicated true altitudes during the glide slope capture.

2. Transition to ILS final.

a. Design steps.

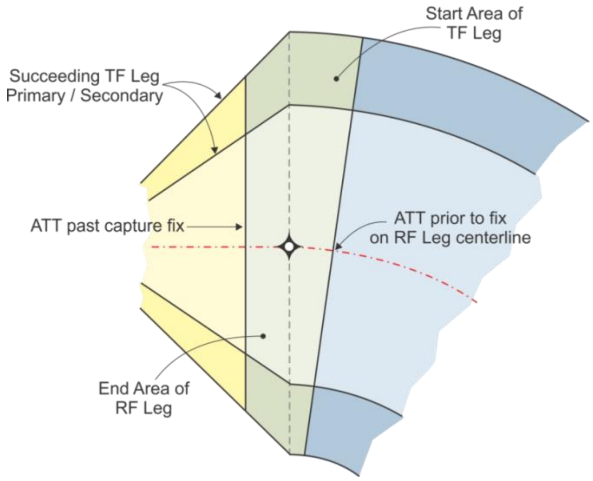
(1) *Step 1*. Evaluate the final, intermediate, and initial segments in accordance with the applicable paragraphs, except:

(a) Establish a capture fix. Construct a TF leg aligned with the FAC that is common to all intermediate segments. The start fix of this leg is designated the capture fix. The preliminary location of the capture fix may be less than 2 NM but no closer than 1 NM prior to the PFAF. The capture fix must be coded as the Final Approach Course Fix (FACF).

(b) Intermediate OEA.

1. RNAV construction [see Chapter 3.]. The segment remains at full width until reaching 2 NM prior to the PFAF or the capture fix, whichever is closer to the PFAF. The primary and secondary boundaries abeam this point taper uniformly to the X and Y OCS outer boundaries abeam the PFAF. The preceding leg construction follows the tapering boundaries until reaching the appropriate ATT beyond the fix [see figure C-1 for RF example].

Figure C-1. Intermediate OEA, RNAV construction



2. RNP AR construction [see Chapter 4.]. Use the same RNP value throughout the entire intermediate construction and verify that the final segment supports the selected RNP with formula C-1. Where the selected RNP is less than the minimum, increase the RNP or decrease the length of final. The capture leg ends at the plotted position of the PFAF.

Formula C-1. Minimum RNP

$$\mathrm{Min_{RNP}} = \frac{0.3048 \times (\mathrm{FAS_{Length}} \times 0.10752 + 678.496)}{1852 \times 2}$$

Where:
$\mathrm{FAS_{Length}}$ = length (feet) of the final approach segment

Example:

$$\mathrm{Min_{RNP}} = \frac{0.3048 \times (50201.07 \times 0.10752 + 678.496)}{1852 \times 2}$$

$$\mathrm{Min_{RNP}} = 0.5$$

(c) *Step 2*. Specify a mandatory altitude at the capture fix which results in as close to a two-degree descent angle as operationally possible [see formula C-2]. Where the descent angle is greater than two degrees, lengthen the segment and/or alter the preceding segment to provide a lower fix altitude.

Formula C-2. Capture Leg Descent Angle

$$\mathrm{Capture_{\theta}} = \mathrm{atan}\left(\frac{\mathrm{Capture_{alt}} - \mathrm{PFAF_{alt}}}{\mathrm{Capture_{Length}}}\right)$$

Where:
$\mathrm{Capture_{alt}}$ = MSL altitude (feet) of the capture fix
$\mathrm{Capture_{Length}}$ = distance (feet) between the capture fix and PFAF

Example:

$$\mathrm{Capture_{\theta}} = \mathrm{atan}\left(\frac{2300 - 2000}{8754.28}\right)$$

$$\mathrm{Capture_{\theta}} \approx 1.96°$$

(d) *Step 3*. Calculate the actual aircraft altitude at the capture fix using the default 40°C ΔISA in formula C-3. Alternatively, determine the average hot temperature using a process similar to paragraph 3-3-4.a(1)(a) and calculate its deviation from ISA using formula 3-3-3.

Formula C-3. High Temperature Capture Fix Altitude

$$\mathrm{High_{alt}} = \frac{(\mathrm{Capture_{alt}} - \mathrm{LTP_{elev}}) \times \Delta\mathrm{ISA_{High}}}{288 + \Delta\mathrm{ISA_{High}} - 0.5 \times 0.00198 \times \mathrm{Capture_{alt}}} + \mathrm{Capture_{alt}}$$

Where:
$\mathrm{Capture_{alt}}$ = MSL altitude (feet) of the capture fix
$\Delta\mathrm{ISA_{High}}$ = 40°C or historical high temperature deviation from ISA

Example:

$$\mathrm{High_{alt}} = \frac{(2300 - 1000) \times 40}{288 + 40 - 0.5 \times 0.00198 \times 2300} + 2300$$
$$\mathrm{High_{alt}} \approx 2459.64 \text{ feet}$$

(e) *Step 4*. Calculate the minimum capture leg length using formula C-4. If the designed length is greater than or equal to the minimum, the design is satisfactory. Otherwise, lengthen the leg and reevaluate if fix altitudes change.

Formula C-4. Minimum Capture Leg Length

$$\mathrm{Min_{Length}} = \frac{\mathrm{r} \times \pi}{180} \times \left(90 - \theta - \mathrm{asin}\left[\frac{\cos(\theta) \times (\mathrm{r} + \mathrm{LTP_{elev}} + \mathrm{TCH})}{\mathrm{r} + \mathrm{High_{alt}}}\right]\right) - \mathrm{FAS_{Length}}$$

Where:
$\mathrm{High_{alt}}$ = calculated high temperature capture fix altitude
$\mathrm{FAS_{Length}}$ = length (feet) of the final approach segment

Example:

$$\mathrm{Min_{Length}} = \frac{\mathrm{r} \times \pi}{180} \times \left(90 - 3 - \mathrm{asin}\left[\frac{\cos(3) \times (\mathrm{r} + 1000 + 45)}{\mathrm{r} + 2459.64}\right]\right) - 18221.16$$
$$\mathrm{Min_{Length}} \approx 8443.89 \text{ feet}$$

THIS PAGE IS INTENTIONALLY LEFT BLANK

Appendix D. Radar Vector SID Implementation

1. Purpose. When establishing a radar vector SID certain prerequisites should be considered, the design should account for radar coverage and operational need.

2. Radar Vector SID authorization. The use a radar vector SID is not authorized when:

a. Radar surveillance is not available.

b. 4-D trajectory management is required (i.e., future NextGen implementation).

3. Radar Vector SID application. A radar vector SID that includes a vector leg between routes (Open SID) should only be used when:

a. The initial leg(s) off the runway requires a specific ground track,

b. The intermediate leg(s) is (are) anticipated to require frequent ATC vectoring, and

c. The final (PBN route) leg(s) require(s) a specific ground track.

Table D-1. Cases for Application of Open SID

	Initial (climb) leg(s) A	**Intermediate (vector) leg B**	**Final (PBN route) leg(s) C**
Case 1	Terrain or obstacles near the initial climb out path	Requires dispersion of ground tracks to distribute noise footprint	Requires a path to avoid terrain or obstacles terrain
Case 2	Noise sensitive areas or congested airspace near the initial climb out path	Frequently requires ATC intervention to separate departing traffic e.g. from hub/satellite airfields	Joins a PBN route (where containment is assumed at the entry point)
Case 3	Special Use Airspace (e.g., Prohibited or restricted area) near the initial climb out path	Requires attaining a minimum altitude that enables a large angle turn to be accomplished (to reduce DP departure track miles)	Requires a path around airspace

Note: If at least one condition from each of the three vertical columns is present, an Open SID may be considered. For example, where 1A, 1B, and 1C are present or if 1A, 2B, and 3C apply, an Open SID is a viable design option.

THIS PAGE IS INTENTIONALLY LEFT BLANK

Appendix E. Geospatial Standard

Section 1. General

1. Algorithms and methods. Algorithms and methods are described for calculating geodetic locations (latitudes and longitudes) on the World Geodetic System of 1984 (WGS-84) ellipsoid for PBN procedure construction. These algorithms utilize existing distance and azimuth methods to compute intersections and tangent points. The methods apply corrections to an initial spherical approximation until the error is less than the maximum allowable error.

2. Algorithm format and definitions. The algorithm prototypes and parameter descriptions are given below using a "C-like" syntax. However, the algorithm steps are described in pseudo-code to maintain clarity and readability. Because this appendix is intended for computer application, formulas are expressed in radians.

a. Geodetic locations. For convenience, a structure called LLPoint is used to represent both latitude and longitude of a geodetic coordinate.

b. Geodesic curves. A geodesic curve is the minimal-length curve connecting two geodetic locations. Since the planar geodesic is a straight line, we will often informally refer to a geodesic as a "line." Geodesics will be represented in data using two LLPoint structures.

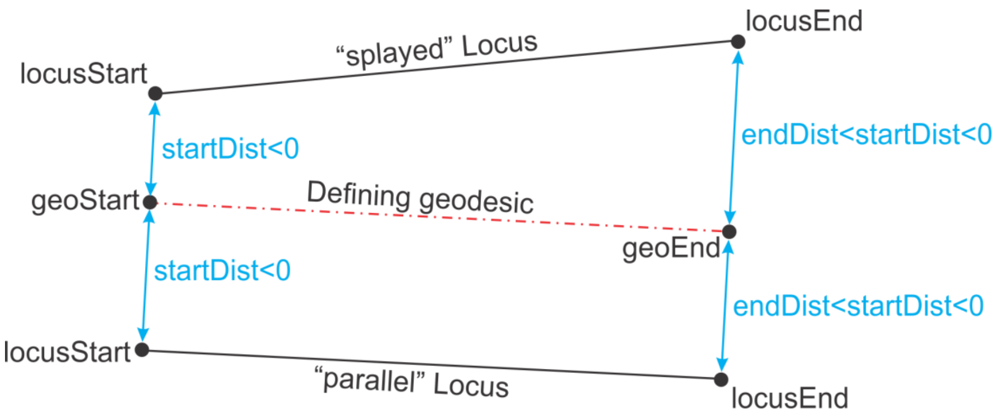
c. Fixed radius arc. A geodetic arc can be defined by a center point and radius distance. The circular arc is then the set (or locus) of points whose distance from the center point is equal to the radius. If an arc subtends an angle of less than 360 degrees, then its start azimuth, end azimuth, and orientation must be specified. The orientation is represented using a value of ±1, with +1 representing a counterclockwise arc and -1 representing a clockwise arc. The distance between the start and end points must be checked. If it is less than a predetermined tolerance value, then the arc will be treated like a complete circle.

d. Locus of points relative to a geodesic. A locus of points relative to a geodesic is the set of all points such that the perpendicular distance from the geodesic is defined by a continuous function w(P) which maps each point P on the geodesic to a real number. For the purposes of procedure design, w(P) will be either a constant value or a linear function of the distance from P to geodesic start point. In the algorithms that follow, a locus of points is represented using the following C structure:

```
typedef struct {
    LLPoint geoStart; /* start point of geodesic */
    LLPoint geoEnd; /* end point of geodesic */
    LLPoint locusStart; /* start point of locus */
    LLPoint locusEnd; /* end point of locus */
    double startDist; /* distance from geodesic to locus at geoStart */
    double endDist; /* distance from geodesic to locus at geoEnd */
    int lineType; /* 0, 1, or 2 */
} Locus;
```

(1) The startDist and endDist parameters define where the locus lies in relation to the defining geodesic. If $(endDist = startDist)$ then the locus will be described as being “parallel” to the geodesic, while if $(endDist \neq startDist)$ then the locus is “splayed.” Furthermore, the sign of the distance parameter determines which side of the geodesic the locus is on. The algorithms described in this appendix assume the following convention: if the distance to the locus is positive, then the locus lies to the right of the geodesic; if the distance is negative, then the locus lies to the left. These directions are relative to the direction of the geodesic as viewed from the geoStart point.

Figure E-1. Two Examples Loci Defined Relative to a Single Geodesic



(2) If memory storage is limited, then either the startDist/endDist or locusStart/ locusEnd elements may be omitted from the structure, since one may be calculated from the other. However, calculating them once upon initialization and then storing them will reduce computation time.

(3) The lineType attribute is used to specify the locus’s extent. If it is set to 0 (zero), then the locus exists only between geoStart and geoEnd. If $(lineType = 1)$ then the locus begins at geoStart but extends beyond geoEnd. If $(lineType = 2)$ then the locus extends beyond both geoStart and geoEnd.

3. Ellipsoidal formulas. Ellipsoidal formulas must be used in evaluating fixes, courses, and distance between fixes.

a. WGS-84 parameters. The semi-minor axis (b) is derived from the semi-major axis (a) and flattening parameters (f) using the relation $\mathrm{b = a \ \times (1 - f)}$.

$$a = 6{,}378{,}137.0 \text{ meters}$$
$$b \approx 6{,}356{,}752.31 \text{ meters}$$
$$f = \frac{1}{298.257223563}$$

b. Direct and inverse algorithms. The Direct and Inverse cases utilize formulae from T. Vincenty’s, Survey Review XXIII, No. 176, April 1975: Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations.

4. Spherical Approximations. An earth radius is needed for spherical approximations. The appropriate radius is the geometric mean of the WGS-84 semi-major and semi-minor axes:

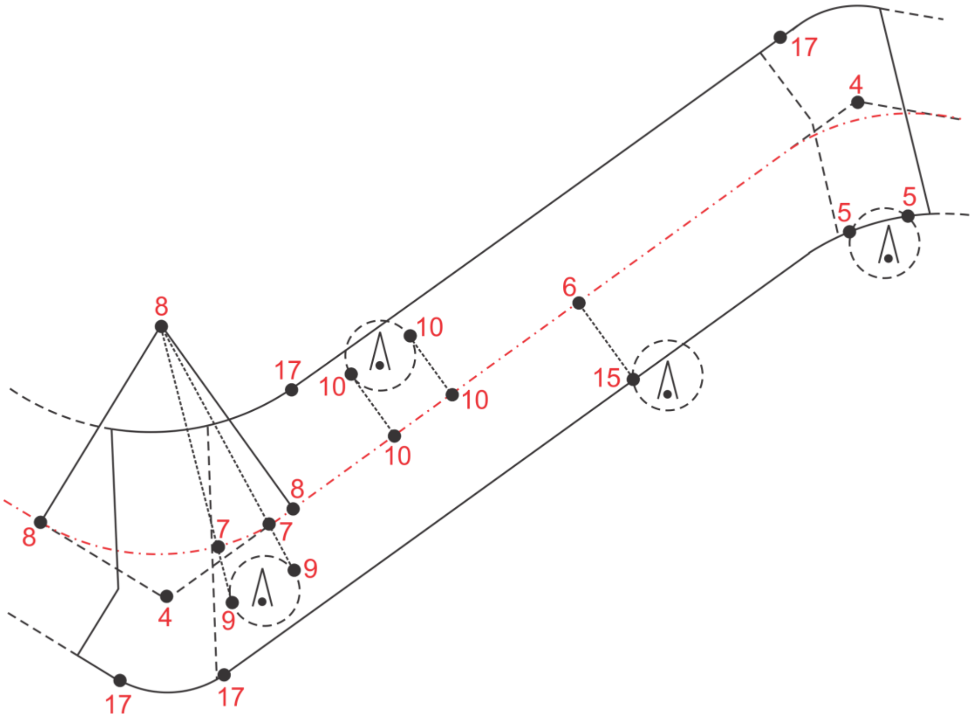
$$SPHERE_{RADIUS} = \sqrt{a \times b}$$

$$SPHERE_{RADIUS} \approx 6{,}367{,}435.68\ meters$$

5. Accuracy. Perform calculations to at least 15 significant digits. For the purpose of determining geodetic positions, perform sufficient iterations to converge within 1 cm in distance and 0.002 arc seconds in bearing. A locus of points must lie within 30 cm of the true locus to be considered compliant.

6. Geodetic processes. The following are required for evaluating PBN procedures. The numbers in figure E-2 refer to the process used to solve for the point.

Figure E-2. Geodetic Processes



a. Process 1. Find the destination latitude and longitude, given starting latitude and longitude as well as distance and starting azimuth (often referred to as the "direct" or "forward" calculation).

b. Process 2. Compute the geodesic arc length between two points, along with the azimuth of the geodesic at either point (often referred to as the "inverse" calculation).

c. Process 3. Given a point on a geodesic, find a second geodesic that is perpendicular to the given geodesic at that point.

d. Process 4. Given two geodesics, find their intersection point(s).

e. Process 5. Given two constant-radius arcs, find their intersection point(s).

f. Process 6. Given a geodesic and a separate point, find the point on the geodesic nearest the given point.

g. Process 7. Given a geodesic and an arc, find their intersection point(s).

h. Process 8. Given two geodesics and a radius value, find the arc of the given radius that is tangent to both geodesics and the points where tangency occurs.

i. Process 9. Given an arc and a point, determine the geodesic(s) tangent to the arc through the point and the point(s) where tangency occurs.

j. Process 10. Given an arc and a geodesic, determine the geodesic(s) that are tangent to the arc and perpendicular to the given geodesic and the point(s) where tangency occurs.

k. Process 11. Compute the length of an arc.

l. Process 12. Determine whether a given point lies on a particular geodesic.

m. Process 13. Determine whether a given point lies on a particular arc.

n. Process 14. Given a geodesic and a locus, find their intersection point.

o. Process 15. Given a fixed-radius arc and a locus, find their intersection point(s).

p. Process 16. Given two loci, find their intersection.

q. Process 17. Given two loci and a radius, find the center of the arc tangent to both loci and the points of tangency.

Section 2. Useful Functions

1. Calculate angular arc extent. When calculating the angle subtended by an arc, one must take into account the possibility that the arc crosses the northern branch cut, where $\mathbf{0° = 360°}$. The following algorithm accounts for this case.

a. Input/Output. double WGS84GetArcExtent(double startCrs, double endCrs, int orientation, double tol) returns a double precision value containing the arc's subtended angle, where the input values are:

(1) double startCrs = Azimuth from center to start point of arc

(2) double endCrs = Azimuth from center to end point of arc

(3) int orientation = Integer that indicates the direction in which the arc is traversed to go from startCrs to endCrs.

(4) orientation = 1 if the arc is traversed counter-clockwise,

(5) orientation = -1 if the arc is traversed clockwise.

(6) double tol = Maximum error allowed in calculations

b. Algorithm steps.

(1) *Step* 1. If $(abs(startCrs - endCrs) < tol)$ then return $2 \times \pi$.

(2) *Step 2*. If $(orientation < 0)$ then orientation is clockwise. Cast the arc into a positive orientation (counter-clockwise) so only one set of calculations is required:

(a) $temp = startCrs$

(b) $startCrs = endCrs$

(c) $endCrs = temp$

(3) *Step 3*. End if.

(4) *Step* 4. If $(startCrs > endCrs)$ then $angle = startCrs – endCrs$.

(5) *Step* 5. Else $angle = 2 \times \pi + startCrs - endCrs$.

(6) *Step 6*. End if.

(7) *Step* 7. If $(orientation < 0)$ then $angle = -angle$.

(8) *Step 8*. Return angle.

2. Converting geodetic latitude/longitude to ECEF coordinates. Geodetic coordinates may be converted to rectilinear ECEF coordinates using the following formulae[1]. Given geodetic latitude ($\boldsymbol{\varphi}$), geodetic longitude ($\boldsymbol{\theta}$), semi-major axis ($\boldsymbol{a}$) and flattening parameter ($\boldsymbol{f}$), calculate the square of the eccentricity $\boldsymbol{e^2 = f \times (2 - f)}$ and the curvature in the prime vertical:

$$N = \frac{a}{\sqrt{1 - e^2 \times \sin(\varphi)^2}}$$

The ECEF coordinates are then:

$$x = N \times \cos(\varphi) \times \cos(\theta)$$
$$y = N \times \cos(\varphi) \times \sin(\theta)$$
$$z = N \times (1 - e^2) \times \sin(\varphi)$$

3. Signed azimuth difference. It is often necessary to calculate the signed angular difference in azimuth between two geodesics at the point where they intersect. The following functions casts the difference between two geodesics into the range $[\boldsymbol{-\pi},\ \boldsymbol{\pi}]$:

$$signedAzimuthDifference(a_1, a_2) = mod(a_1 - a_2 + \pi, 2\pi) - \pi$$

a. This function returns the angle between the two geodesics as if the geodesic that is oriented along azimuth a_1 were on the positive *x*-axis and the geodesic oriented along azimuth a_2 passed through the origin. In other words, if $signedAzimuthDifference(a_1 a_2) > 0$ azimuth a_2 is to the left when standing at the geodesics' intersection point and facing in the direction of azimuth a_1.

b. The mod function in the definition of signedAzimuthDifference must always return a non-negative value.

Note: The C language's built in fmod function does not have this behavior, so a replacement must be supplied. The following code suffices:

```
double mod(double a, double b) {
a = fmod(a,b);
  if (a < 0.0) a = a + b;
return a;  }
```

4. Approximate fixed radius arc length. Section 3 algorithm 7 describes a method for computing the length of an arc to high precision. The following algorithm provides a solution accurate to one centimeter for an arc whose radius is less than about 300 nautical miles (NM). This algorithm approximates the ellipsoid at the center of the arc in question with a "best fit" sphere, whose radius is computed as the geometric mean of the meridional and prime-vertical curvatures at the arc's center.

a. Given the arc center's latitude (θ), the ellipsoidal semi-major axis (a) and flattening (f), compute the local radius of curvature R as follows:

[1] Dana, Peter H., "Coordinate Conversion Geodetic Latitude, Longitude, and Height to ECEF, X, Y, Z>", 11 February, 2003

$$e^2 = f \times (2 - f)$$

$$M = \frac{a \times (1 - e^2)}{[1 - e^2 \times \sin(\theta)^2]^{\frac{3}{2}}}$$

$$N = \frac{a}{\sqrt{1 - e^2 \times \sin(\theta)^2}}$$

$$R = \sqrt{M \times N}$$

b. If the radius and subtended angle of the of the constant radius arc are r and A, respectively, then the length of the arc is given by:

$$L = A \times R \times \sin\left(\frac{r}{R}\right)$$

Section 3. Basic Calculations

1. Iterative approach. For most of the intersection and projection methods listed below, an initial approximation is iteratively improved until the calculated error is less than the required accuracy. The iterative schemes employ a basic secant method, relying upon a linear approximation of the error as a function of one adjustable parameter. To begin the iteration, two starting solutions are found and used to initialize a pair of two-element arrays.

a. The first array stores the two most recent values of the parameter being adjusted in the solution search. This array is named distarray when the search parameter is the distance from a known point. It is named crsarray when the search parameter is an angle measured against the azimuth of a known geodesic.

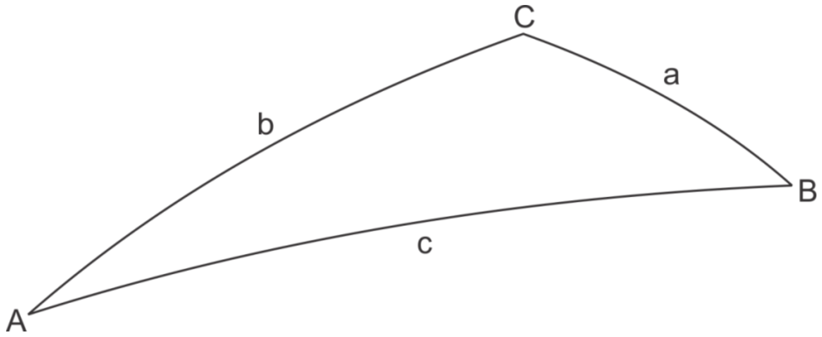
b. The second array (named errarray in the algorithms below) stores the error values corresponding to the two most recent parameter values. Thus, these arrays store a linear representation of the error function. The next solution in each iteration is found by solving for the root of that linear function using the findLinearRoot function. This function returns the value of the search parameter for which the linear error approximation is zero. The returned root is used as the next value in the adjustable parameter and the corresponding error value is calculated. Then the parameter and error arrays are updated and another new root is found.

```
static double findLinearRoot(double* x, double* y, long* err) {
if (x[0] == x[1]) {
    /* function has duplicate x values, no root */
    return x[0];
}
else if (y[0] == y[1]) {
    if (y[0]*y[1] == 0.0) return x[0];
    /* duplicate y values in root function */
    return 0.5*(x[0]+x[1]);
}
    return -y[0]*(x[1]-x[0])/(y[1]-y[0]) + x[0]
}
```

2. Starting solutions. Starting solutions must be provided to start iterating toward a precise solution. Initial solutions may be found in all cases by using spherical triangles to approximate the geodetic curves being analyzed, and then solve for unknown distance and azimuth values using spherical trigonometry formulas.

a. Spherical direction intersection. Given two points A and B and two bearings A to C and B to C, find C.

Figure E-3. Spherical Direction Intersection



(1) *Step 1*. Run Inverse to find arc length from A to B and bearings A to B and B to A. Compute differences of bearings to find angles A and B of the spherical triangle ABC. More than one valid solution may result. Choose the solution closest to the original points.

(2) *Step 2*. Apply the spherical triangle formulas to find the angle C and arc lengths from A to C and from B to C:

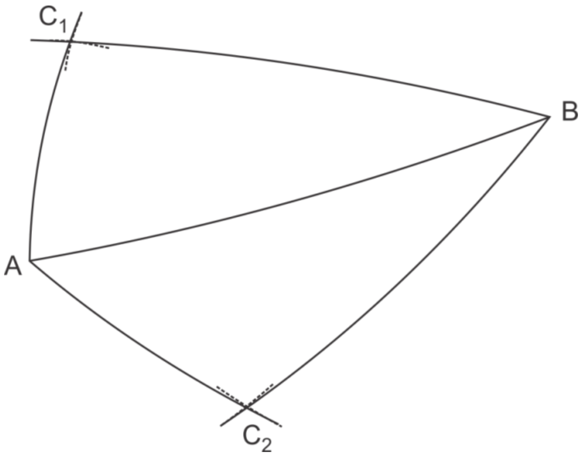
$$C = \cos^{-1}\left[-\cos(A) \times \cos(B) + \sin(A) \times \sin(B) \times \cos\left(\frac{c}{R}\right)\right]$$
$$a = R \times \cos^{-1}\left[\frac{\cos(A) + \cos(B) \times \cos(C)}{\sin(B) \times \sin(C)}\right]$$
$$b = R \times \cos^{-1}\left[\frac{\cos(B) + \cos(A) \times \cos(C)}{\sin(A) \times \sin(C)}\right]$$

Note: If distances a or b result from a reciprocal bearing, assign appropriate negative sign(s).

(3) *Step 3*. Run Direct from A to find C. Use given bearing and computed length b.

b. Spherical distance intersection. Given A, B and distances AC and BC, find C_1 and C_2.

Figure E-4. Spherical Distance Intersection



(1) *Step 1*. Run Inverse to find length and bearings between A and B. Use spherical triangles to find angles $A = BAC_1 = BAC_2, B = ABC_1 = ABC_2, and\ C = BC_1A = BC_2A$:

$$A = \cos^{-1}\left[\frac{\cos\left(\frac{a}{R}\right) - \cos\left(\frac{b}{R}\right) \times \cos\left(\frac{c}{R}\right)}{\sin\left(\frac{b}{R}\right) \times \sin\left(\frac{c}{R}\right)}\right]$$

$$B = \cos^{-1}\left[\frac{\cos\left(\frac{b}{R}\right) - \cos\left(\frac{a}{R}\right) \times \cos\left(\frac{c}{R}\right)}{\sin\left(\frac{a}{R}\right) \times \sin\left(\frac{c}{R}\right)}\right]$$

$$C = \cos^{-1}\left[\frac{\cos\left(\frac{c}{R}\right) - \cos\left(\frac{a}{R}\right) \times \cos\left(\frac{b}{R}\right)}{\sin\left(\frac{a}{R}\right) \times \sin\left(\frac{b}{R}\right)}\right]$$

(2) *Step 2*. Run Direct from A to find C_1 and C_2.

(a) To compute the bearing from A to $C_{1,}$ start with the bearing from A to B and subtract angle A.

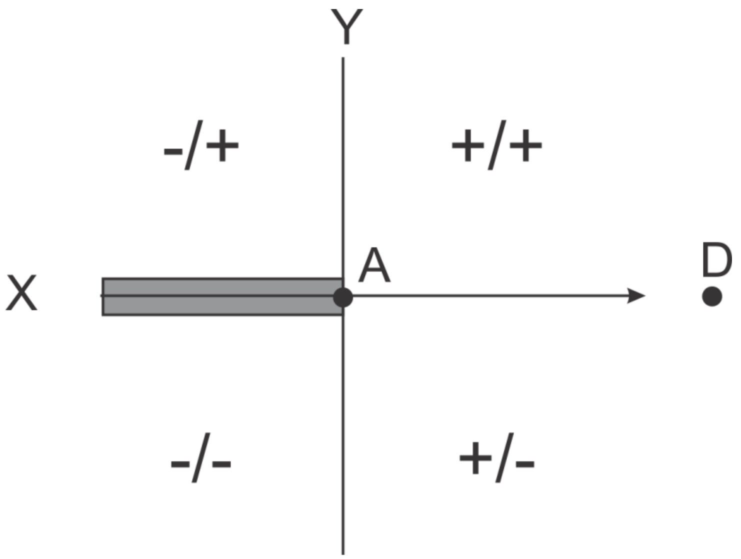
(b) To compute the bearing from A to $C_{2,}$ start with the bearing from A to B and add angle A.

(3) *Step 3*. Use Inverse and spherical triangle formulas to get remaining bearings.

c. Spherical tangent point.

(1) Distance signs. In both cases of the tangent point, distances are signed according figure E-5, where the arrow indicates the bearing from the first point A to the target point D.

Figure E-5. Distance Signs



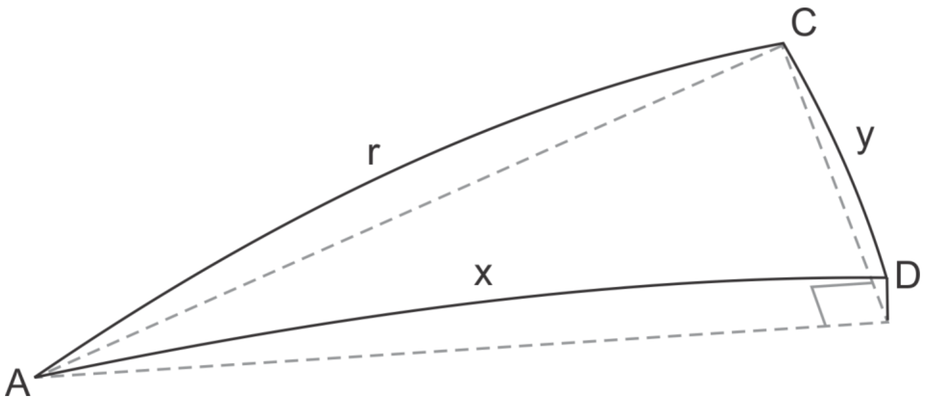
(2) Right spherical triangle. Calculate lengths y and x of the right spherical triangle with the following formulas:

$$y = R \times \sin^{-1}\left[\sin\left(\frac{r}{R}\right) \times \sin(A)\right]$$

$$x = R \times \cos^{-1}\left[\frac{\cos\left(\frac{r}{R}\right)}{\cos\left(\frac{y}{R}\right)}\right]$$

(3) Given two points and a bearing. Given two points, A and C, and a bearing from the first point (A), find the point D along the given bearing extended which is closest to C.

Figure E-6. Two Points and a Bearing



(a) *Step 1*. Run Inverse to find length and bearings between A and C.

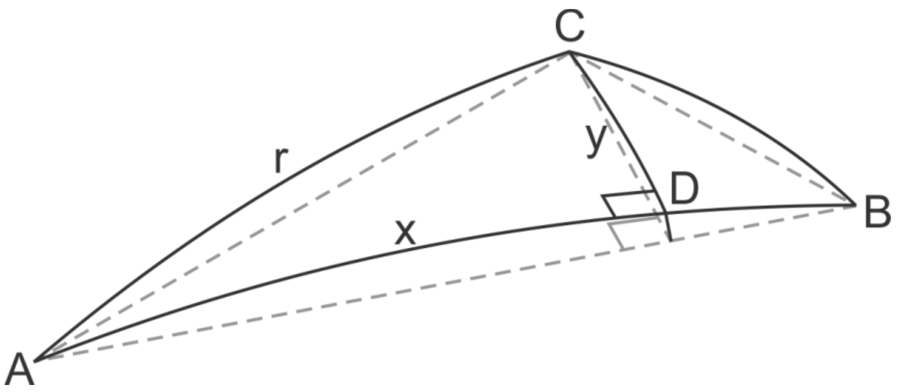
1. Find difference in bearings to compute angle A.

2. Use right spherical triangles to calculate y and x.

(b) *Step 2*. Run Direct from A to find D using given bearing and computed length x.

(4) Given three points. Given three points (A, B, C), find the point (D) on the geodesic line from the first two points which is the perpendicular foot from the third point.

Figure E-7. Three Points



(a) *Step 1*. Use Inverse to determine bearing from A to B.

(b) *Step 2*. Use Inverse to determine bearing and length from A to C.

(c) *Step 3*. Find the difference in bearings to determine angle A.

(d) *Step 4*. Use right spherical triangles to calculate y and x.

(e) *Step 5*. Use Direct to calculate D from A using the computed bearing from A to B and computed distance x.

3. Tolerances. Two different convergence tolerances must be supplied so that the algorithms cease iterating once the error becomes sufficiently small.

a. Forward and inverse. The first tolerance parameter is used in the forward and inverse routines; it is referred to as eps in the algorithm descriptions. Empirical studies have shown that $eps = 0.5 \times e^{-13}$ works well.

b. Intersection and projection. The second parameter, labeled tol, is used in the intersection and projection routines to limit the overall error in the solution. Since the intersection and projection routines make multiple calls to the forward and inverse algorithms, the eps parameter should be several orders of magnitude smaller than the tol parameter to ensure that the iteration methods return correct results. Empirical studies have shown that $tol = 1.0 \times e^{-9}$ works well.

c. Maximum iterations. A maximum iteration count must be supplied to ensure that no algorithms can remain in an infinite loop if convergence is not reached. This parameter can be set by the programmer, but should be greater than five to ensure that all of the algorithms can reach convergence.

4. Geodesic oriented at specified angle. In TERPS procedure design, it is often required to find a geodesic that lies at a prescribed angle to another geodesic. For instance, the end lines of an obstacle evaluation area (OEA) are typically projected from the flight path at a prescribed angle. Since the azimuth of a geodesic varies over the length of the curve, the angle between two geodesics must be measured by comparing the azimuth of each geodesic at the point where they intersect. The following pseudo-code represents an algorithm that will calculate the correct azimuth at any point on a geodesic described by its start and end points. This azimuth can easily be extended to find the azimuth of an intersecting geodesic at the point if the angle of intersection is known.

a. Input/Output. double WGS84GeodesicCrsAtPoint(LLPoint startPt, LLPoint endPt, LLPoint testPt, int length, double* startCrs, double* revCrs, double* distToPt, long* err, double tol, double eps) returns a double representing the azimuth of the intersecting geodesic, where the inputs are:

(1) LLPoint startPt = Coordinates of start point of given geodesic

(2) LLPoint endPt = Coordinates of end point of geodesic

(3) LLPoint testPt = Point at which course of geodesic is to be determined

(4) double* startCrs = Azimuth of geodesic at startPt in radians

(5) double* revCrs = Reciprocal azimuth of geodesic at endPt in radians

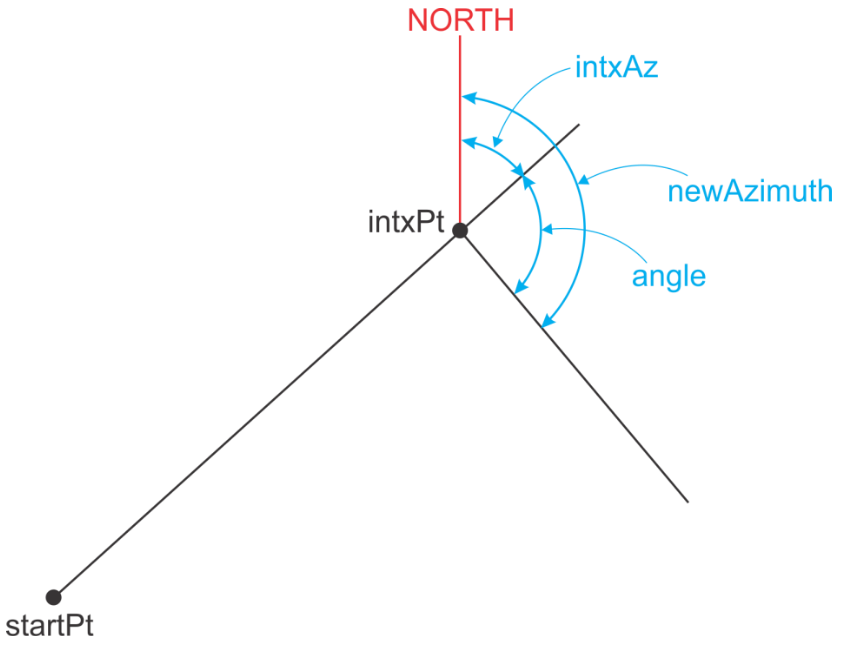
(6) double* distToPt = Distance from startPt to testPt in NM

(7) double tol = Accuracy tolerance for intersection calculation

(8) double eps = Convergence parameter for forward/inverse algorithms

b. Algorithm steps.

Figure E-8. Projecting a Geodesic through a Point along the Specified Azimuth



(1) *Step 1*. Use the WGS84PtIsOnLine algorithm to check that testPt actually lies on geodesic defined by startPt and endPt.

(2) *Step* 2. Use Inverse algorithm to determine course and distance from testPt to startPt. Denote course as crsToStart.

(3) *Step 3*. Use Inverse algorithm to determine course and distance from testPt to endPt. Denote course as crsToEnd.

(4) *Step 4*. If testPt lies on geodesic between startPt and endPt, then the correct azimuth is crsToEnd.

(5) *Step 5*. If testPt lies on the geodesic beyond endPt, then the correct azimuth is $crsStart + \pi$.

(6) *Step 6*. Return the calculated azimuth.

Note: If an angle is positive, then the new geodesic will lie to the right of the given geodesic (from the perspective of standing at the start point and facing toward the end point); otherwise, the new geodesic will lie to the left.

5. Determine if point lies on geodesic. This algorithm returns a true value if a point lies on and within the bounds of a given geodesic. The bounds of the geodesic are specified by two pieces of information: the end point coordinates and an integer length code. If the length code is set to 0, then the geodesic is understood to exist only between its start and end points, so a value of true will be returned only if the test point also lies between the start and end points. If the length code is set to 1, then the geodesic is understood to extend beyond its end point to a distance of one half of earth's circumference from its end point. If the length code is set to 2, then the geodesic is understood to extend beyond both the start and end points. This algorithm relies on the concept of equality for two LLPoint structures. This will be defined to mean that the distance between the two LLPoints, as calculated using the inverse algorithm, is less than tol.

a. Input/Output. int WGS84PtIsOnLine(LLPoint startPt, LLPoint endPt, LLPoint testPt, LineType lengthCode, double tol, double eps) returns an integer value indicating whether testPt lies on geodesic, where the inputs are:

(1) LLPoint startPt = Geodetic coordinate of line start point

(2) LLPoint endPt = Geodetic coordinate of line end point

(3) LLPoint testPt = Geodetic coordinate of point to test

(4) LineType lengthCode = Integer that specifies extent of line.

(a) 0: geodesic exists only between startPt and endPt.

(b) 1: geodesic extends beyond endPt.

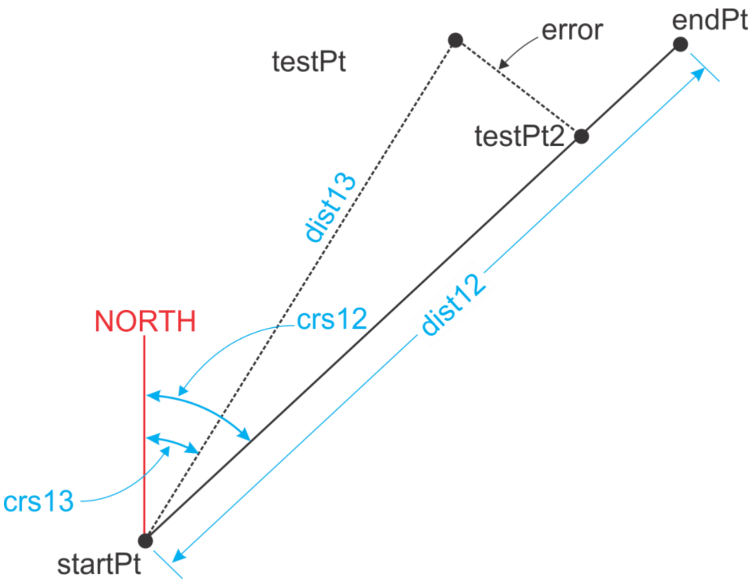
(c) 2: geodesic extends behind startPt.

(5) double tol = Maximum difference allowed in distance

(6) double eps = Convergence parameter for forward/inverse algorithms

b. Algorithm steps.

Figure E-9. Entities for Testing Whether a Point Lies on a Geodesic



(1) *Step 1*. Use inverse algorithm to calculate the azimuth and distance from startPt to endPt. Denote these values by crs12 and dist12, respectively.

(2) *Step 2*. Use WGS84PtIsOnCrs algorithm to determine if testPt lies on geodesic given by startPt and endPt:

(a) Use inverse algorithm to calculate the distance from startPt to testPt. Denote this value by tmpDist1Test.

(b) Use direct algorithm to project a point from startPt, along crs12, a distance equal to tmpDist1Test. Denote this point by comparePt.

(c) Use WGS84PointsAreSame algorithm to determine if testPt is equal to comparePt.

(3) *Step 3*. Examine error to determine whether testPt lies on the geodesic within tol as follows:

(a) If $(error \leq tol)$ then,

1. If $(lengthCode > 0)$ or $(dist13 - dist12 \leq tol)$ then $onLine = true$.

2. Else $onLine = false$.

(b) End if.

(c) If ($lengthCode = 2$),

1. Use the direct algorithm to project point from startPt, along $crs12 + \pi$ a distance dist13. Again, denote this point again by testPt2.

2. Use the inverse algorithm to recalculate error, which is the distance from testPt to testPt2.

3. If ($error \leq tol$) then $onLine = true$, Else $onLine = false$.

(d) End if.

6. Determine if point lies on arc. This algorithm returns a non-zero (true) value if the sample point lies on and between the bounds of the given arc. The arc is defined by its center point, radius, start azimuth, end azimuth, and orientation. A positive orientation parameter indicates that the arc is traversed in a counterclockwise sense, while a negative orientation parameter indicates that the arc is traversed clockwise. This algorithm is used in conjunction with the arc intersection functions (section 5 algorithms 2, 3, and 6) to determine whether the computed intersections lie within the bounds of the desired arc.

a. Input/Output. int WGS84PtIsOnArc(LLPoint center, double radius, double startCrs, double endCrs, ArcDirection orientation, LLPoint testPt, double tol) returns an integer value indicating whether testPt lies on arc, where the inputs are:

(1) LLPoint center = Geodetic coordinates of arc center

(2) double radius = Arc radius

(3) double startCrs = True azimuth from center to start of arc

(4) double endCrs = True azimuth from center to end of arc

(5) ArcDirection orientation = Orientation of the arc [+1 for counter-clockwise; -1 for clockwise]

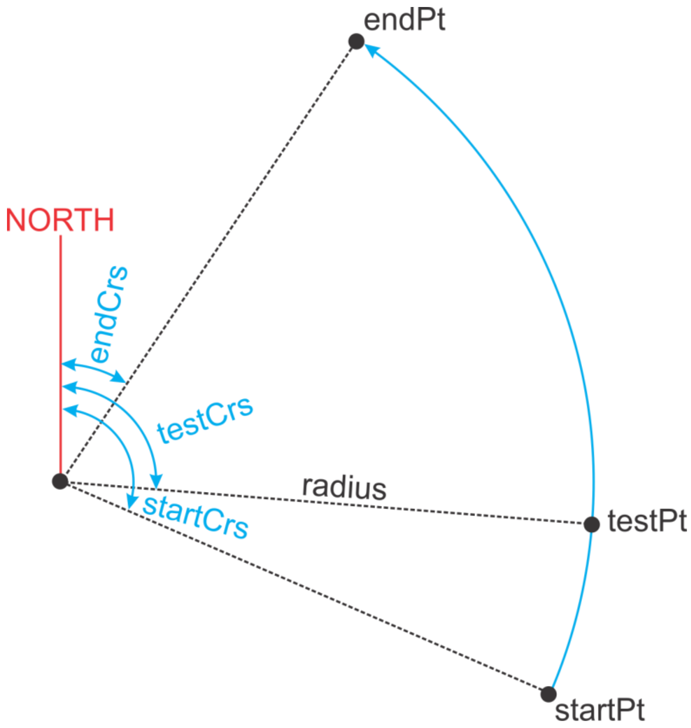
(6) LLPoint testPt = Geodetic coordinate of point to test

(7) double tol = Maximum error allowed in solution

(8) double eps = Convergence parameter for forward/inverse algorithms

b. Algorithm steps.

Figure E-10. Entities for Testing Whether a Point Lies on an Arc



(1) *Step 1*. Use inverse algorithm to calculate distance and azimuth from center to testPt. Denote values as dist and crs, respectively.

(2) *Step 2*. If $(abs(dist - radius) > tol)$ then testPt is not correct distance from center, onArc = false.

(3) *Step 3*. Else

(a) Use section 2 algorithm 1, Calculate Angular Arc Extent, to calculate the angle subtended by the full arc. Denote this value by arcExtent.

(b) If $(arcExtent = 360°)$ then $onArc = true$.

(c) Else

<u>1.</u> Use the inverse algorithm to calculate the azimuth from center to testPt. Denote this value by testCrs.

<u>2.</u> Use section 2 algorithm 1, Calculate Angular Arc Extent, to calculate the angle subtended by and arc starting at startCrs, but ending at testCrs, with the same orientation. Denote this value by subExtent.

<u>3.</u> If $(-.002 \leq subExtent \leq arcExtent + .002)$ then traversing arc from startCrs to endCrs, one would encounter testPt, so it must lie on arc, $onArc = true$.

(d) End if.

(4) *Step 4*. End if.

7. Calculate length of fixed radius arc. A fixed radius arc on an ellipsoid does not generally lie in a plane. Therefore, the length of the arc cannot be computed using the usual formula for the circumference of a circle. The following algorithm takes the approach of dividing the arc into many sub-arcs. Three points are then calculated on each sub-arc. Since any three points in space uniquely determine both a plane and an arc, the three points on each sub-arc are used to calculate the radius and subtended angle of the planar arc that contains all three points. The length of the approximating planar arc is then calculated for each sub-arc. The sum of the sub-arc lengths approaches the length of the original arc as the number of sub-arc increases (and each sub-arc's length decreases). A simpler method that is sufficiently accurate for arcs with radius less than about 300 NM is described in section 2 algorithm 4.

a. Input/Output. double WGS84DiscretizedArcLength (LLPoint center, double radius, double startCrs, double endCrs, int orient, int *n, double tol) returns a double precision value representing the length of the arc, where the inputs are:

(1) LLPoint center = Geodetic coordinates of arc center

(2) double radius = Arc radius

(3) double startCrs = True azimuth from center to start of arc

(4) double endCrs = True azimuth from center to end of arc

(5) int orient = Orientation of the arc [+1 for counter-clockwise; -1 for clockwise]

(6) int *n = Reference to integer used to return number of steps in discretized arc

(7) double tol = Maximum allowed error

(8) double eps = Convergence parameter for forward/inverse algorithms

b. Algorithm steps.

Figure E-11. Calculating Arc Length

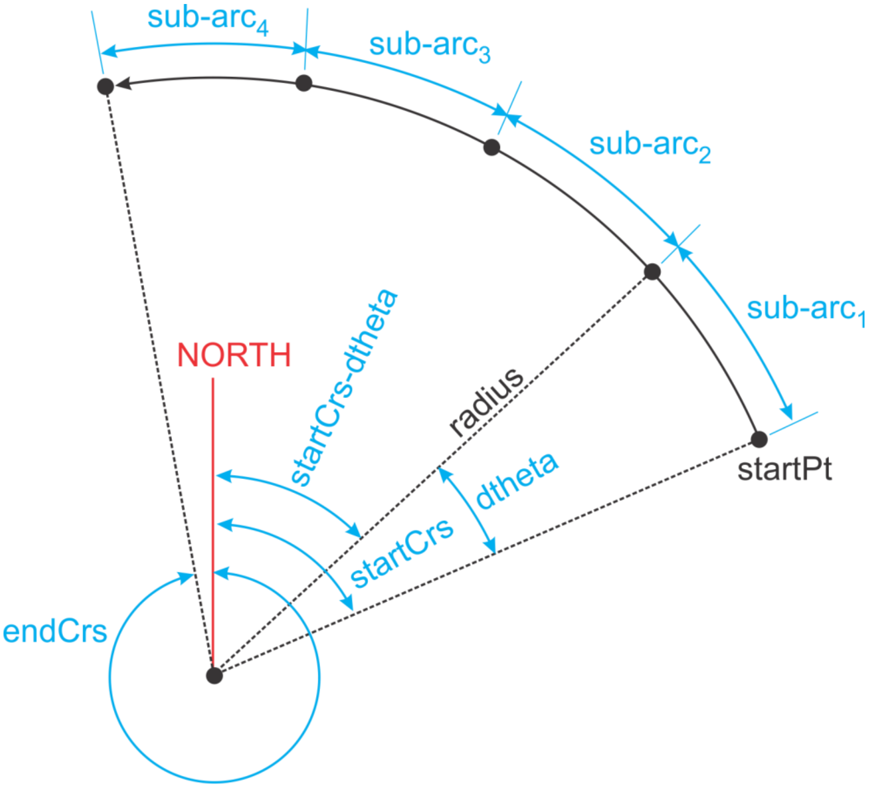
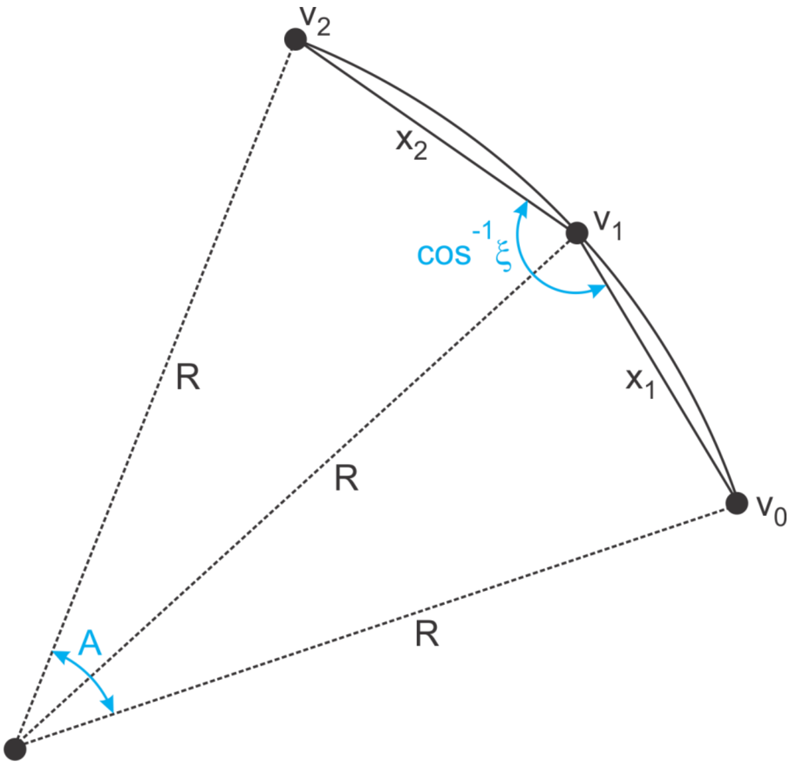


Figure E-12. Calculating the Sub-Arc Length



(1) *Step 1*. Set initial number of sub-arcs to use. The fixed value $n = 16$ has been found through trial-and-error to be a good starting value. Alternatively, the initial value of n may be calculated based on the arc's subtended angle and its radius (i.e., its approximate arc length).

(2) *Step 2*. Convert center point to Earth-Centered, Earth-Fixed (ECEF) coordinates, v0 according to section 2 algorithm 2.

(3) *Step 3*. Compute subtended angle, subtAngle, using section 2 algorithm 1.

(4) *Step 4*. Set iteration count, $k = 0$.

(5) *Step* 5. Do while $k = 0$ or $(error > tol)$ and $(k \leq maximumIterationCount)$

(a) Calculate subtended angle of each sub-arc, $dtheta = \frac{subtAngle}{n}$.

(b) Use direct algorithm from center, using startCrs and distance radius, to project start point of arc. Denote this point by p1.

(c) Convert p1 to ECEF coordinates. Denote this vector by v1.

(d) Initialize $arcLength = 0$.

(e) For $i = 0$ to n:

1. Compute azimuth from arc center to end point of sub-arc number i: $theta \ = startCrs + i \times dtheta$.

2. Use direct algorithm from center, using azimuth $theta + 0.5 \times dtheta$ and distance radius, to project middle point of sub-arc. Denote this point by p2.

3. Convert p2 to ECEF coordinate v2.

4. Use direct algorithm from center, using azimuth $theta + dtheta$ and distance radius, to project endpoint of sub-arc. Denote this point by p3.

5. Convert p3 to ECEF coordinate v3.

6. Subtract v2 from v1 to find chord vector between p1 and p2. Denote this vector by chord1. Compute $x1 = |chord1|$.

7. Subtract v2 from v3 to find chord vector between p3 and p2. Denote this vector by chord2. Compute $x2 = |chord2|$.

8. Compute dot product of chord1 and chord2. Denote this value as d.

9. Use the following calculation to compute the length L of the sub-arc:

a. $xi = \frac{d}{x1 \times x2}$

b. $sigma = \sqrt{1 - xi^2}$

c. $R = \frac{x2 \times \sqrt{\left(\frac{x1}{x2} - xi\right)^2 + sigma^2}}{2 \times sigma}$

d. $A = 2 \times [\pi - acos(xi)]$

e. $L = R \times A$

$1\ \xi = \frac{d}{x_1 \times x_2}$

$2\ \sigma = \sqrt{1 - \xi^2}$

$3\ R = \frac{x_e \times \sqrt{\left(\frac{x_1}{x_2 - \xi}\right)^2 + \sigma^2}}{2 \times \sigma}$

$4\ A = 2 \times [\pi - \cos^{-1}(\xi)]$

Note: Since the arc length is a planar (not geodetic) calculation, the subtended angle A is not equal to dtheta.

10. Add L to cumulative length to get total length of sub-arcs through sub-arc number i: $length = length + L$.

(f) End for loop.

(g) Compute error, which is the change in length calculation between this iteration and the last: $error = abs(length - oldLength)$.

(h) Increment the iteration count: $k = k + 1$.

(i) Double the number of sub-arcs: $n = 2 \times n$.

(j) Save the current length for comparison with the next iteration: $oldLength = length$.

(6) *Step 6*. End while loop.

(7) *Step 7*. Return length.

8. Find distance from defining geodesic to locus. When computing a position on a locus of points, it is necessary to solve for the distance from the defining geodesic to the locus. This distance is constant if the locus is designed to be “parallel” to the defining geodesic. However, it is necessary to allow the locus distance to vary linearly with distance along the geodesic, since in some cases the locus will splay away from the defining geodesic. To account for this, we have included startDist and endDist attributes in the Locus structure defined above. For a given point on the geodesic (or given distance from the geodesic start point), the distance to the locus can then be calculated. The two algorithms described below carry out the computation of locus distance for different input parameters. If the distance from the geodesic start point to the point of interest is known, then WGS84DistToLocusD may be used to calculate the locus distance. If instead a point on the defining geodesic is given, the WGS84DistToLocusP may be used. The latter algorithm simply computes the distance from the geodesic start point to the given point and then invokes the former algorithm. Therefore, steps are described for WGS84DistToLocusD only.

a. Input/Output.

(1) double WGS84DistToLocusD (Locus loc, double distance) returns the distance from the defining geodesic to the locus at the given distance from loc.geoStart, where the inputs are:

(a) Locus loc = Locus of interest

(b) double distance = Distance from locus start point to point where distance is to be computed

(2) double WGS84DistToLocusP (Locus loc, LLPoint geoPt, double *faz, double tol, double eps) returns the distance from the defining geodesic to the locus at the given point, where the inputs are:

(a) Locus loc = Locus of interest

(b) LLPoint geoPt = Point on defining geodesic

(c) double *faz = Pointer used to return forward azimuth of geodesic at geopt. This is needed if geopt is not between geoStart and geoEnd.

(d) double tol = Maximum allowable error

(e) double eps = Convergence parameter for forward/inverse algorithm

b. Algorithm steps. The following steps are followed if the distance from loc.geoStart is given. If a point on the geodesic (geoPt) is given instead, then first use the inverse algorithm to compute the distance from geoPt to loc.geoStart and then follow the following steps (note that distance must be signed negative if the locus's line type is 2 and geoPt is farther from geoEnd than it is from geoStart):

(1) *Step 1*. Use the inverse function to compute the length of the locus's defining geodesic. Denote this value as geoLen.

(2) *Step* 2. If $(geoLen = 0)$ then $distToLoc = 0.0$.

(3) *Step 3*. Else.

$$distToLoc = loc.startDist + \frac{distance}{geoLen} \times (loc.endDist - loc.startDist)$$

(4) *Step 4*. End if.

(5) *Step 5*. Return distToLoc.

9. Determine if point lies on locus. This algorithm compares the position of a given point with the position of the corresponding point on the locus. The corresponding point on the locus is found by projecting the given point onto the locus's defining geodesic curve, computing the correct distance from there to the locus, and then projecting a point at that distance perpendicular to the geodesic. If distance from the corresponding point to the given point is less than the error tolerance, then a reference to the projected point on the geodesic is returned. Otherwise a null reference is returned. An alternative implementation could simply return true or false, rather than references. However, it is more efficient to return the projected point as this is often needed in subsequent calculations.

a. Input/Output. int WGS84PtIsOnLocus (Locus loc, LLPoint testPt, LLPoint* ptOnGeo, double tol, double eps) returns a reference to the projection of testPt on the locus's defining geodesic if testPt lies on the locus and NULL otherwise, where the inputs are:

(1) Locus loc = Locus of Interest

(2) LLPoint testPt = Point to test against locus

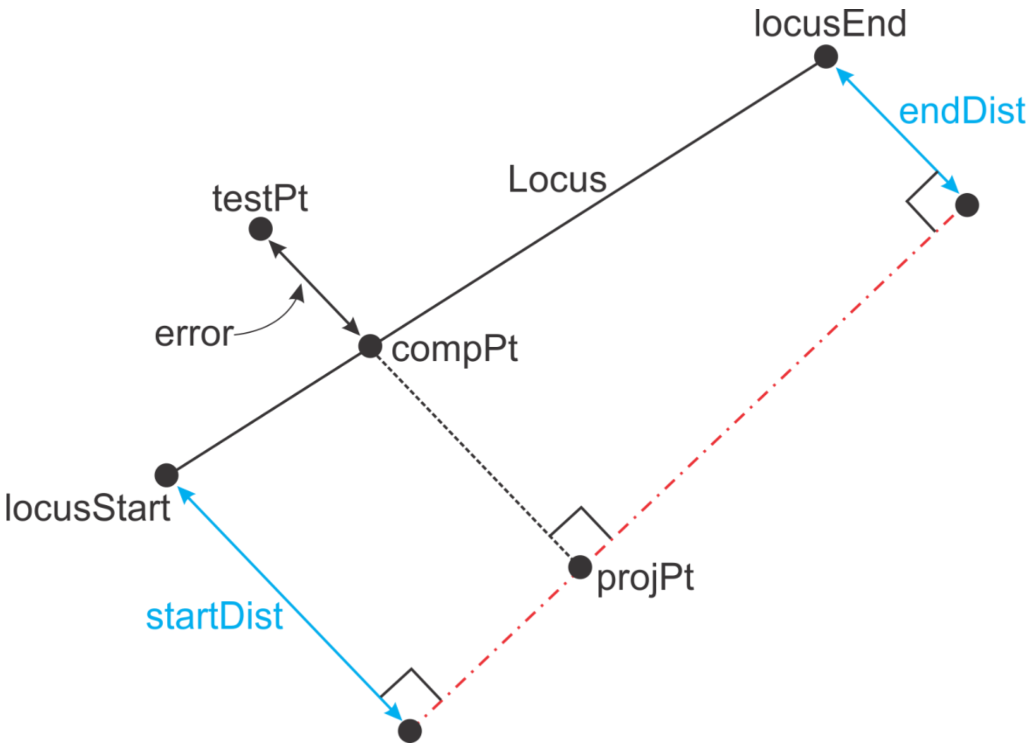
(3) LLPoint* ptOnGeo = Pointer to LLPoint, updated with point on defining geodesic abeam the given point on the locus.

(4) double tol = Maximum allowable error

(5) double eps = Convergence parameter for forward/inverse algorithms

b. Algorithm steps.

Figure E-13. Locating a Point Relative to a Locus



(1) *Step 1*. If testPt is the same as loc.geoStart or loc.geoEnd then return a reference to ptOnGeo containing the appropriate point.

(2) *Step 2*. Use section 4 algorithm 1 to project testPt onto the locus's defining geodesic. Denote the projected point as ptOnGeo.

(3) *Step 3*. Use section 3 algorithm 5 to determine whether ptOnGeo lies on the locus's defining geodesic. This will account for an infinite or semi-infinite locus. If it does not, then return 0 (false).

(4) *Step 4*. Use the inverse algorithm to find the course between loc.geoStart and testPt. Use this course to determine which side of the locus testPt falls. Apply the appropriate sign to this distance, distFromPoint.

(5) *Step 5*. Use section 3 algorithm 8 to calculate the correct expected locus distance, locDist.

(6) *Step 6*. If $(abs(distFromPoint - locDist) \leq tol)$ then the point is on the locus. Return a reference to the projection on the defining geodesic.

10. Compute course of locus. This algorithm is analogous to the inverse algorithm for a geodesic. It is used by other locus algorithms when the direction of the locus is needed.

a. Input/Output. double WGS84LocusCrsAtPoint (Locus loc, LLPoint testPt, LLPoint* geoPt, double* perpCrs, double tol) returns the course of the locus at the given point. Also sets values of calculation byproducts, including the corresponding point on the locus's geodesic and the course from the given point toward the geodesic point, where the inputs are:

(1) Locus loc = Locus of Interest

(2) LLPoint testPt = Point at which course will be calculated

(3) LLPoint* geoPt = Projection of testPt on defining geodesic

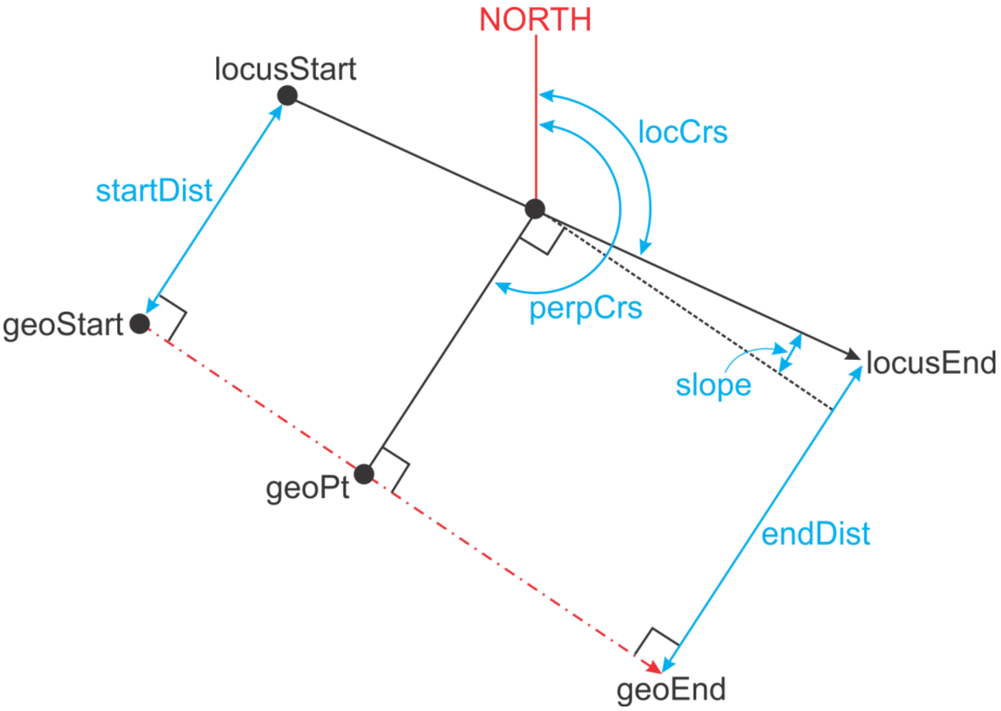
(4) double* perpCrs = Course for testPt to geoPt

(5) double tol = Maximum allowable error

(6) double eps = Convergence parameter for forward/inverse algorithms

b. Algorithm steps.

Figure E-14. Angle Used to Calculate the Course of a Locus



(1) *Step 1*. Use section 3 algorithm 9 to determine whether testPt lies on loc. This same step will return a reference to the projection of testPt onto the defining geodesic. Denote this reference as geoPt.

(2) *Step 2*. If $(geoPt = NULL)$ then testPt is not a valid point at which to calculate the locus's course. Return -1.0. (Valid course values are in the range $[0, 2 \times \pi]$.)

(3) *Step 3*. Use the inverse algorithm to calculate the course and distance from testPt to geoPt, denoted by perpCrs and perpDist, respectively.

(4) *Step 4*. Use section 3 algorithm 8 to calculate distToLoc, the distance from the geodesic to the locus at geoPt. This step is required to determine which side of the geodesic the locus lies on because perpDist will always be positive.

(5) *Step 5*. Calculate the slope of the locus relative to the geodesic:

$$slope = \frac{loc.endDist - loc.startDist}{geoLen}$$

(6) *Step 6*. Convert the slope to angular measure in radians:

$$slope = \text{atan}(\text{slope})$$

(7) *Step 7*. Adjust the value of the perpendicular course by slope. This accounts for how the locus is approaching or receding from the geodesic: $perpCrs = perpCrs + slope$

(8) *Step 8*. If (distToLoc < 0) then testPt lies to the left of the geodesic, so perpCrs points to the right of the locus's course: $locCrs = perpCrs - \frac{\pi}{2}$

(9) *Step 9*. Else testPt lies to the right of the geodesic so perpCrs points to the left of the locus's course: $locCrs = perpCrs + \frac{\pi}{2}$

(10) *Step 10*. Return locCrs.

Section 4. Projections

1. Project point to geodesic. This algorithm is used to determine the shortest distance from a point to a geodesic. It also locates the point on the geodesic that is nearest the given point.

a. Input/Output. long WGS84PerpIntercept (LLPoint pt1, double crs12, LLPoint* pt2, LLPoint pt3, double* crsFromPoint, double* distFromPoint, double tol) returns a reference to an LLPoint structure that contains the coordinates of the projected point, where the inputs are:

(1) LLPoint pt1 = Coordinates of geodesic start point

(2) double crs13 = Initial azimuth of geodesic at start point

(3) LLPoint pt3 = Coordinates of point to be projected to geodesic

(4) LLPoint* pt2 = Reference to LLPoint that will be updated with coordinates of projected point.

(5) double* crsFromPoint = Reference to azimuth of geodesic from pt3 to projected point, in radians.

(6) double* distFromPoint = Reference to distance from pt3 to projected point, in radians.

(7) double tol = Maximum error allowed in solution

(8) double eps = Convergence parameter for forward/inverse algorithms

b. Algorithm steps. This algorithm treats the geodesic as unbounded, so that projected points that lie "behind" the geodesic starting point pt1 will be returned. If it is desired to limit solutions to those that lie along the forward direction of the given geodesic, then *Step 4(g)* may be modified to return a NULL solution.

Figure E-15. Projecting a Point to a Geodesic

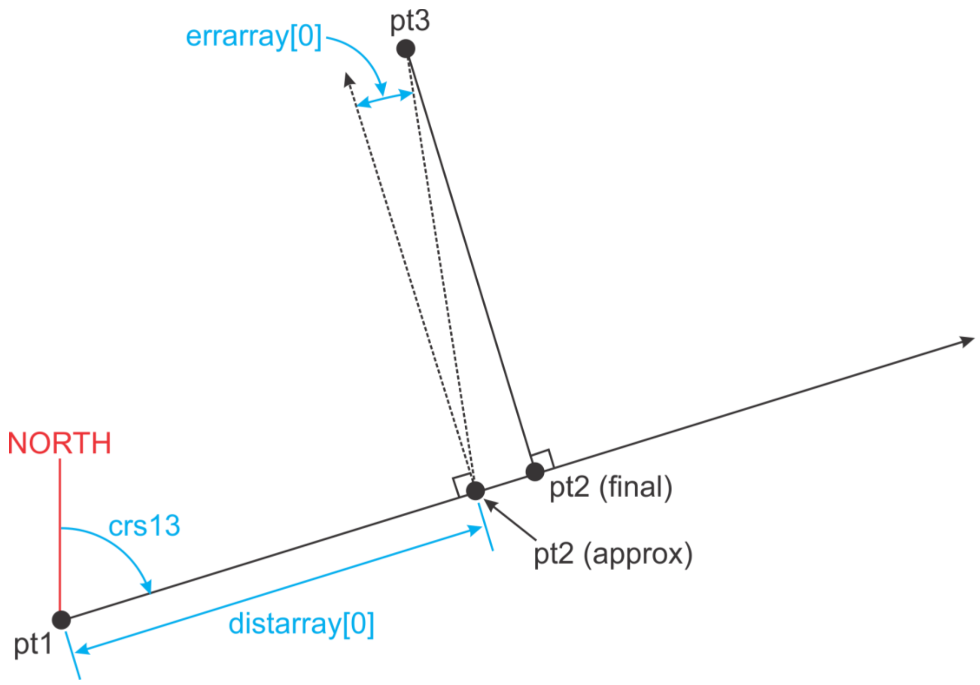
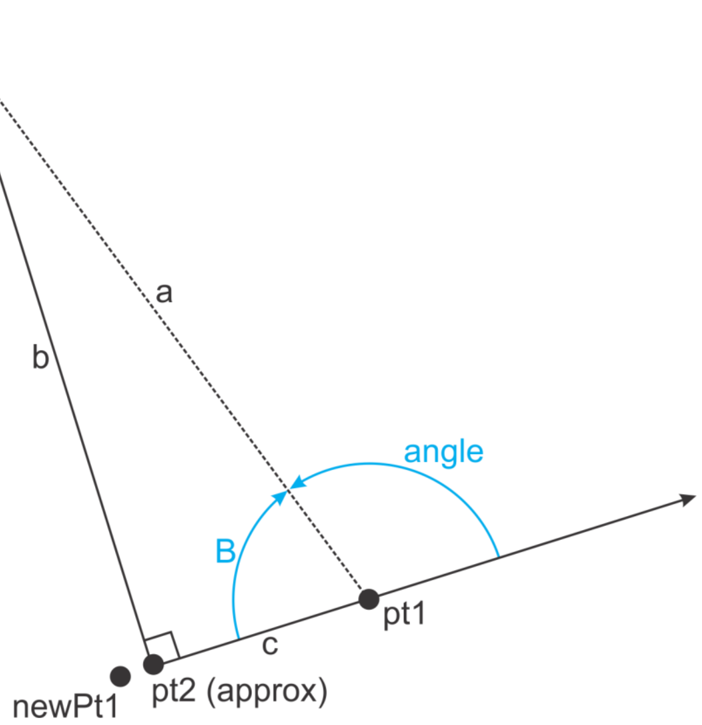


Figure E-16. Elements of Spherical Triangle Used to Determine New Geodesic Starting Point When Projected Point Lies Behind Given Starting Point



(1) *Step 1*. Use the inverse algorithm to calculate the distance, azimuth, and reverse azimuth from pt1 to pt3. Denote these values as crs13, crs31, and dist13, respectively.

(2) *Step 2*. Calculate the angle between the given geodesic and the geodesic between pt1 and pt3. This is accomplished using signedAzimuthDifference function [see section 2 algorithm 3]: $angle = abs(signedAzimuthDifference(crs13, crs12))$

(3) *Step 3*. If $(dist13 \leq tol)$ then pt2 is the same point as pt1.

(4) *Step 4*. If $(\frac{\pi}{2} - angle < tol)$ then the projected point pt2 is very close to or behind pt1 (the start of the geodesic), so extend the geodesic backward far enough to catch the projection. Use a spherical triangle approximation to calculate the needed extension distance:

(a) $B = angle$

(b) $a = \frac{dist13}{SPHERE_{RADIUS}}$

(c) $b = \text{asin}[\sin(B) \times \sin(a)]$

(d) $dist12 = 2 \times SPHERE_{RADIUS} \times \text{atan}(\tan[0.5 \times (a - b)]) \times \sin[0.5 \times (A - B)]$

(e) If $(abs(dist12) < tol)$ then the projected point is identical to pt1 to within the required accuracy:

<u>1.</u> $crsFromPoint = crs31$

<u>2.</u> $distFromPoint = dist13$

<u>3.</u> $Return\, pt2 = pt1$

(f) End if.

(g) Use the direct algorithm to move pt1 along reverse geodesic course. Use $1.1 \times dist12$ for the distance, $crs12 + \pi$ for the azimuth, and then store the new location in the temporary variable newPt1. A distance greater than dist12 is used to compensate for possible errors in the spherical approximation.

(h) Use the inverse algorithm to calculate the azimuth from newPt to pt1. This value replaces the original azimuth value crs12, rename newPt1 as pt1: $pt1 = newPt1$.

(5) *Step 5*. Calculate the approximate distance from pt1 to the projected point using the spherical triangle formula from *Steps 4(a)* through *4(d)*. Denote the approximate distance found as dist13.

(6) *Step 6*. Use the direct algorithm to project a point on the given geodesic distance dist13 from pt1. Use pt1 for the starting point, dist12 for distance, and crs12 for azimuth. Denote the computed point by pt2.

(7) *Step 7*. Use the inverse algorithm to calculate the azimuth crs21 from pt2 to pt1.

(8) *Step 8*. Use the inverse algorithm to calculate the azimuth crs23 and distance dist23 from pt3 to pt2.

(9) *Step 9*. Calculate the angle between the geodesics that intersect at pt3, and cast that angle into the range $[0, \pi]$ using the following formula:

$$angle = abs\big(signedAzimuthDifference(crs21, crs23)\big)$$

(10) *Step 10*. Calculate the error and store it as the first element in the error function array: $errarray[0] = angle - \frac{\pi}{2}$.

(11) *Step 11*. Store the current distance from pt1 to pt2 in the distance function array: $distarray[0] = dist12$.

(12) *Step 12*. A second distance/error value must be calculated before linear interpolation may be used to improve the solution. The following formula may be used: $distarray[1] = distarray[0] + errarray[0] \times dist23$.

(13) *Step 13*. Use the direct algorithm to project point on the given geodesic distance distarray[1] from pt1. Use pt1 for the starting point, distarray[1] for distance, and crs12 for azimuth. Denote the computed point by pt2.

(14) *Step 14*. Use the inverse algorithm to calculate the azimuth crs21 from pt2 to pt1.

(15) *Step 15*. Use the inverse algorithm to calculate the azimuth crs23 from pt2 to pt3.

(16) *Step 16*. Calculate the error in angle:

$$errarray[1] = abs\big(signedAzimuthDifference(crs21, crs23)\big) - \frac{\pi}{2}$$

(17) *Step 17*. Initialize the iteration count: $k = 0$.

(18) *Step* 18. Do while $(k = 0)$ or $(error > tol)$ and $(k < maxIterationCount)$

(a) Use linear approximation to find root of errarray as a function of distarray. This gives an improved approximation to dist12.

(b) Use the direct algorithm to project point on the given geodesic distance dist12 from pt1. Use pt1 for the starting point, dist12 for distance, and crs12 for azimuth. Denote the computed point by pt2.

(c) Use the inverse algorithm to calculate the azimuth crs21 from pt2 to pt1.

(d) Use the inverse algorithm to calculate the distance dist23, azimuth crs32, and reverse azimuth crs23 from pt3 to pt2.

(e) Update distarray and errarray with the new values:

1. $distarray[0] = distarray[1]$

2. $errarray[0] = errarray[1]$

3. $distarray[1] = dist13$

4. $errarray[1] = abs(signedAzimuthDifference(crs21, crs23)) - \frac{\pi}{2}$

(f) Calculate the difference between the two latest distance values. This serves as the error function for measuring convergence: $error = abs(distarray[1] - distarray[0])$.

(19) *Step 19*. End while.

(20) *Step* 20. Set $crsToPoint = crs32$.

(21) *Step* 21. Set $distToPoint = dist23$.

(22) *Step 22*. Return pt2.

2. Project point to locus from point on defining geodesic. Given a point on the defining geodesic, this algorithm computes the corresponding point on the locus.

a. Input/Output. LLPoint WGS84PtOnLocusP (Locus loc, LLPoint geoPt, LLPoint* ptonloc, double* perpCrs, double tol, double eps) returns the point on the locus that is abeam the given point, where the inputs are:

(1) Locus loc = Locus of Interest

(2) LLPoint geoPt = Point on defining geodesic

(3) LLPoint* ptonloc = Pointer to LLPoint, updated with coordinates of point on locus abeam given point.

(4) double* perpCrs = Pointer to double, updated with azimuth from point on geodesic to point on locus.

(5) double tol = Maximum allowable error

(6) double eps = Convergence parameter for forward/inverse algorithms

b. Algorithm steps.

(1) *Step 1*. Use section 3 algorithm 8 (with point input) to determine the distance from geoPt to the locus. Denote this distance as distp.

(2) *Step 2*. If $(distp = 0)$ then return geoPt.

(3) *Step 3*. Use the inverse algorithm to compute the course from geoPt to the start point of the defining geodesic. Denote this value as fcrs.

(4) *Step 4*. If ($distp > 0.0$) then the locus lies to the right of the geodesic. Let

$$*\,perpCrs = fcrs + \frac{\pi}{2}$$

(5) *Step 5*. Else the locus lies to the left of the geodesic. Let $*\,perpCrs = fcrs\ - \frac{\pi}{2}$.

(6) *Step 6*. End if.

(7) *Step 7*. Use the direct algorithm to project a point along *perpCrs, distance abs(distp) from geoPt. Denote the point as ptonLoc.

(8) *Step 8*. Return ptonLoc.

3. Project point to locus nearest given point. This algorithm returns the point on a locus nearest the given sample point.

a. Input/Output. LLPoint* WGS84LocusPerpIntercept(Locus loc, LLPoint pt2, double* crsFromPoint, double* distFromPoint, double tol) returns a reference to an LLPoint structure that contains the coordinates of the projected point, where the inputs are:

(1) Locus loc = Locus structure to which point will be projected

(2) LLPoint pt2 = Coordinates of point to be projected to locus

(3) double* crsFromPoint = Reference to value that will store the course from pt2 to projected point

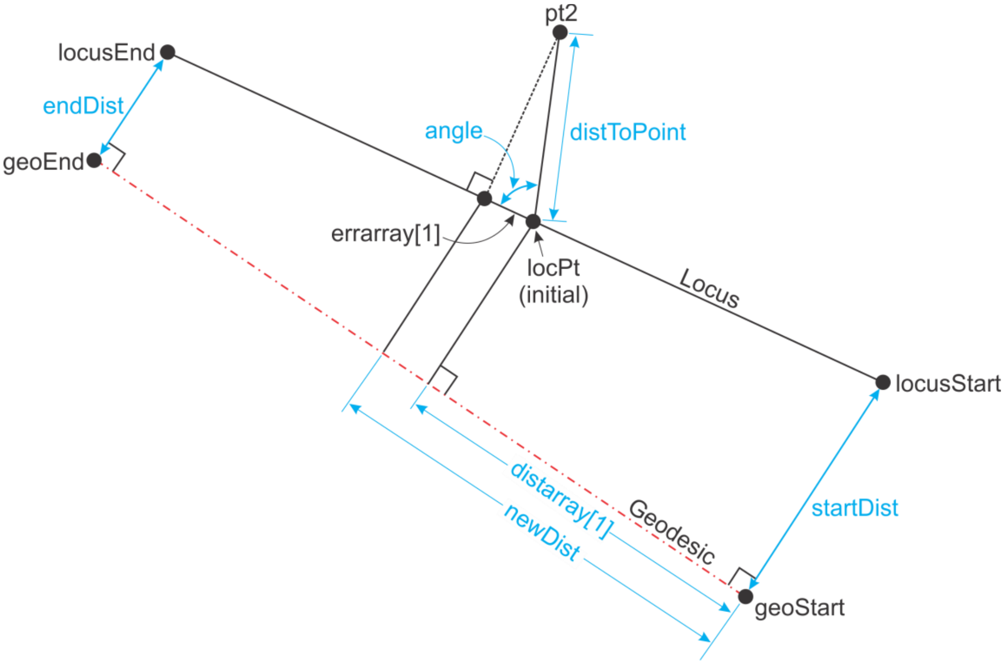
(4) double* distFromPoint = Reference to value that will store the distance from pt2 to projected point

(5) double tol = Maximum error allowed in solution

(6) double eps = Convergence parameter for forward/inverse algorithms

b. Algorithm steps.

Figure E-17. Projecting a Point to a Locus



(1) *Step 1*. Define the course and distance from loc.geoStart to loc.geoEnd as gcrs and gdist, respectively. This course and distance is a part of the locus structure:

(a) $gcrs = loc.geoAz$

(b) $gdist = loc.geoLength$

(2) *Step 2*. If $(abs(loc.startDist - loc.endDist) \; < \; tol)$ then the locus is "parallel" to its defining geodesic. In this case, the projected point on the locus will lie on the geodesic joining pt2 with its projection on the defining geodesic, and the calculation is simplified:

(a) Apply section 4 algorithm 1 to project pt2 onto the defining geodesic of loc. Use loc.geoStart, gcrs, and pt2 as input parameters. The intersection point, perpPt, will be returned along with the course and distance from pt2 to perpPt. Denote the course and distance values as crsFromPoint and distFromPoint, respectively.

(b) Use section 4 algorithm 2 to project a point locPt on the locus from perpPt on the geodesic.

(c) Use the inverse algorithm to recalculate distFromPoint as the distance between pt2 and locPt.

(d) Return locPt.

(3) *Step 3*. End if.

(4) *Step 4*. Use the inverse algorithm to compute lcrs, the course from loc.locusStart to loc.locusEnd.

(5) *Step 5*. Use section 4 algorithm 1 to project pt2 onto the geodesic approximation of the locus. Pass loc.locusStart, lcrs, and pt2 as parameters. Denote the computed point as locPt. (In general, this point will not exactly lie on the locus. We will adjust its position so that it is on the locus in a subsequent step.)

(6) *Step 6*. Calculate the locus inclination angle, relative to its geodesic:

$$locAngle = \operatorname{atan}\left(\frac{loc.startDist - loc.endDist}{gdist}\right)$$

(7) *Step 7*. Use section 4 algorithm 1 to project locPt onto the locus's defining geodesic. Pass loc.geoStart, gcrs, and locPt as parameters. Denote the computed point as geoPt.

(8) *Step 8*. Use the inverse function to calculate the distance from loc.geoStart to geoPt. Store this value as distarray[1].

(9) *Step 9*. Initialize the iteration count: $k = 0$

(10) *Step 10*. Do while $(k = 0)$ or $abs(errarray[1] > tol)$ and $(k < maxIterationCount)$

(a) Use section 4 algorithm 2 with distarray[1] to project a point onto the locus. Reassign locPt as this point.

(b) Use section 3 algorithm 10 to recompute lcrs, the course of the locus at locPt.

(c) Use the inverse algorithm to compute crsToPoint and distToPoint, the course and distance from locPt to pt2.

(d) Compute the signed angle between the locus and the geodesic from locPt to pt2: $angle = signedAzimuthDifference(lcrs, crsToPoint)$

(e) Store the approximate error as: $errarray[1] = -distToPoint \times \cos(angle)$. This converts the error in angle into an error in distance which can be compared to tol.

(f) If $(k = 0)$ then a direct calculation is used to improve the approximation: $newDist = distarray[1] + errarray[1] \times \cos(locAngle)$

(g) Else use a linear root finder with distarray and errarray to solve for the distance value that makes the error zero. Denote this value as newDist.

(h) End if.

(i) Update the distance and error arrays:

1. $distarray[0] = distarray[1]$

2. $errarray[0] = errarray[1]$

3. $distarray[1 \ = newDist$

(11) *Step 11*. End while.

(12) *Step 12*. Return locPt.

4. Tangent projection from point to arc. This projection is used in obstacle evaluation when finding the point on an RF leg or fly-by turn path where the distance to an obstacle must be measured.

a. Input/Output. long WGS84PointToArcTangents(LLPoint point, LLPoint center, double radius, LLPointPair tanPt, int* n, double tol) returns a reference to an LLPoint structure that contains the coordinates of the points where geodesics through point are tangent to arc, where the inputs are:

(1) LLPoint point = Point from which lines will be tangent to arc

(2) LLPoint center = Geodetic centerpoint coordinates of arc

(3) double radius = Radius of arc

(4) LLPointPair tanPt = Two-element array of LLPoint objects that will be updated with tangent points' coordinates

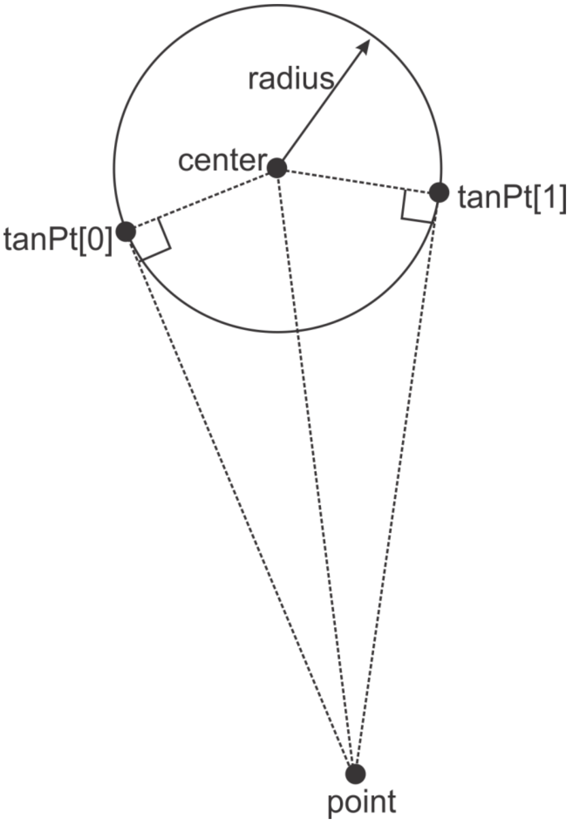
(5) int* n = Reference to number of tangent points found (0, 1, or 2)

(6) double tol = Maximum error allowed in solution

(7) double eps = Convergence parameter for forward/inverse algorithms

b. Algorithm steps. This algorithm treats the arc as a complete circle, so either zero or two tangent points will be returned. If the arc is bounded and two tangent points are found, then each point must be tested using section 3 algorithm 6 to determine whether they lie within the arc’s bounds.

Figure E-18. Projecting Point to Tangent Points on an Arc



(1) *Step 1*. Use the inverse algorithm to calculate the distance, azimuth, and reverse azimuth from point to center. Denote these values by crsToCenter, crsFromCenter, and distToCenter, respectively.

(2) *Step 2*. If $(abs(distToCenter - radius) < tol)$ then point lies on the arc and is a tangent point:

(a) Set $n = 1$.

(b) Return $tanPt = point$.

(3) *Step 3*. Else if $(distToCenter < radius)$ then point lies inside of the arc and no tangent points exist, return no solution.

(4) *Step 4*. End if.

(5) *Step 5*. There must be two tangent points on the circle, so set $n = 2$.

(6) *Step 6*. Use spherical trigonometry to compute approximate tangent points:

(a) $a = \frac{distToCenter}{SPHERE_{RADIUS}}$

(b) $b = \frac{radius}{SPHERE_{RADIUS}}$

(c) $C = \text{acos}\left[\frac{\tan(b)}{\tan(a)}\right]$. This is the approximate angle between the geodesic that joins point with center and the geodesic that joins center with either tangent point.

(7) *Step 7*. Initialize iteration count $k = 0$.

(8) *Step 8*. Do while ($k = 0$) or ($abs(error) > tol$) and ($k < maxIterationCount$):

(a) Use the direct algorithm to locate tanPt[0] on arc. Use center as the starting point, radius as the distance, and $courseFromCenter + C$ as the azimuth.

(b) Use the inverse algorithm to calculate the azimuth from tanPt[0] to center. Denote this value as radCrs.

(c) Use the inverse algorithm to calculate the azimuth from tanPt[0] to point. Denote this value as tanCrs.

(d) Use the function in section 2 algorithm 3 to calculate the angle between the two courses and cast it into the range $[-\pi, \ \pi]$:
$diff = signedAzimuthDifference(radCrs, taxCrs)$

(e) Compute the error $error = abs(diff) - \frac{\pi}{2}$.

(f) Adjust the value of C to improve the approximation $C = C + error$.

(g) Increment the iteration count $k = k + 1$.

(9) *Step 9*. End while loop.

(10) *Step 10*. Repeat *Steps 7-9* to solve for tanPt[1]. In each iteration; however, use crsFromPoint–C for azimuth in *Step 8(a).*

(11) *Step 11*. Return tanPt[0] and tanPt[1].

5. Project arc to geodesic. This algorithm is used for obstacle evaluation when finding a point on the straight portion of TF leg where distance to an obstacle must be measured.

a. Input/Output. long WGS84PerpTangentPoints(LLPoint lineStart, double crs, LLPoint center, double radius, LLPointPair linePts, LLPointPair tanPts, double tol) updates geodesic intercepts, but returns no output, where input values are:

(1) LLPoint lineStart = Start point of geodesic to which arc tangent points will be projected

(2) double crs = Initial course of geodesic

(3) LLPoint center = Geodetic coordinates of arc center

(4) double radius = Arc radius

(5) LLPointPair linePts = Two-element array of projected points on Geodesic

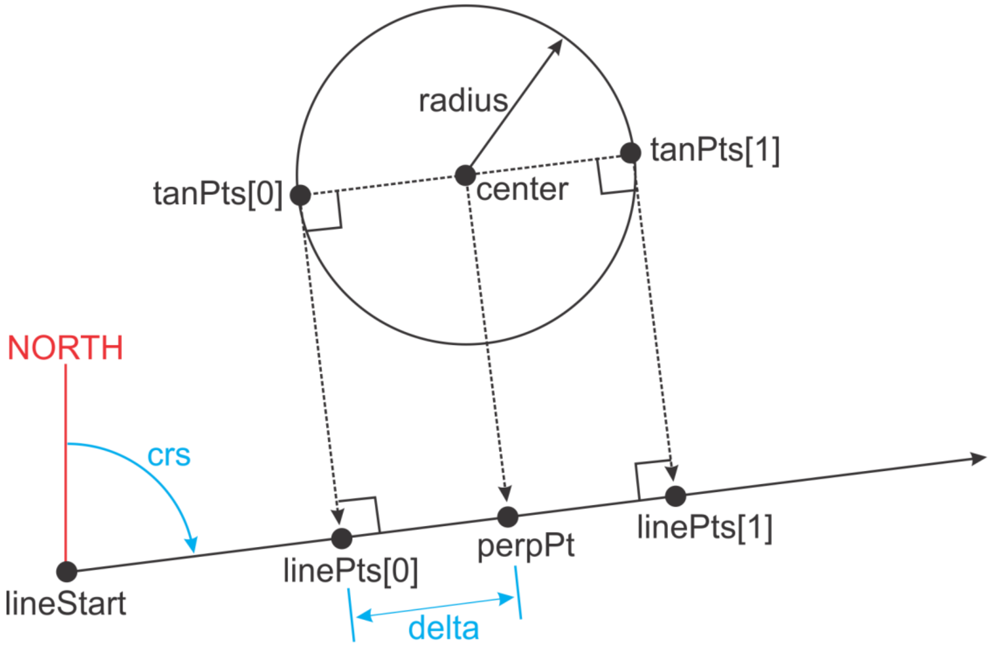
(6) LLPointPair tanPts = Two-element array of tangent points on arc

(7) double tol = Maximum error allowed in solution

(8) double eps = Convergence parameter for forward/inverse algorithms

b. Algorithm steps.

Figure E-19. Projecting an Arc to a Geodesic



(1) *Step 1*. Use the inverse algorithm to calculate the distance, azimuth, and reverse azimuth from lineStart to center. Denote these values as distStartToCenter, crsStartToCenter, and crsCenterToStart, respectively.

(2) *Step 2*. Compute the angle between the given geodesic and the geodesic that joins lineStart to center [see section 2 algorithm 3]:
$angle1 = signedAzimuthDifference(crs, crsStartToCenter)$

(3) *Step 3*. If $(abs(distStartToCenter \times (crsStartToCenter - crs)) < tol)$ then center lies on the given geodesic, which is a diameter of the circle. In this case, the tangent points and project points are the same:

(a) Use the direct algorithm to compute tanPts[0]. Use lineStart as the starting point, crs as the azimuth, and $distStartToCenter - radius$ as the distance.

(b) Use the direct algorithm to compute tanPts[0]. Use lineStart as the starting point, crs as the azimuth, and $distStartToCenter + radius$ as the distance.

(c) Set $linePts[0] = tanPts[0]$.

(d) Set $linePts[1] = tanPts[1]$.

(e) Return all four points.

(4) *Step 4*. End if.

(5) *Step 5*. Use section 4 algorithm 1 to project center to the geodesic defined by lineStart and crs. Denote the projected point by perpPt.

(6) *Step 6*. Use the inverse algorithm to calculate the distance, azimuth, and reverse azimuth from perpPt to lineStart. Denote these values by dist12 and crs21, respectively.

(7) *Step 7*. Set $delta = radius$.

(8) *Step 8*. Initialize iteration count: $k = 0$.

(9) *Step 9*. Do while ($k = 0$) or ($abs(error) > tol$) and ($k < maxIterationCount$):

(a) Use the direct algorithm to compute linePts[0]. Use perpPt as the starting point, delta as the distance, and $crs21 + \pi$ as the azimuth.

(b) Use the inverse algorithm to calculate the course from linePts[0] to perpPt. Denote this value by strCrs.

(c) Calculate the azimuth, perpCrs, from linePts[0] to the desired position of tanPts[0]. The azimuth depends upon which side of the line the circle lies, which is given by the sign of angle1:

1. If the circle lies to the right of the line: $perpCrs = strCrs + \frac{\pi}{2}$.

2. If the circle lies to the left of the line: $perpCrs = strCrs - \frac{\pi}{2}$.

(d) Use section 4 algorithm 1 to project center onto the geodesic passing through linePts[0] at azimuth perpCrs. Section 4 algorithm 1 will return the projected point, tanPts[0], along with the distance from center to tanPts[0]. Denote this distance by radDist.

(e) Calculate the error, the amount that radDist differs from radius: $error = radDist - radius$.

(f) Adjust the distance from lineStart to linePts[0]: $delta = delta - error$.

(g) Increment the iteration count: $k = k + 1$.

(10) *Step 10*. End while loop.

(11) *Step 11*. Repeat *Steps 7-10* to solve for linePts[1] and tanPts[1]. In each iteration; however, use crs21 for azimuth in *Step 9(a)*. Using the final delta value for the first iteration in the search for linePts[1] will make the code more efficient (i.e., don’t repeat *Step 7*).

(12) *Step 12*. Return linePts[0], linePts[1], tanPts[0], and tanPts[1].

Section 5. Intersections

1. Intersection of two geodesics. The following algorithm computes the coordinates where two geodesic curves intersect. Each geodesic is defined by its starting coordinates and azimuth at that coordinate. The algorithm returns a single set of coordinates if the geodesics intersect and returns a null solution (no coordinates) if they do not.

a. Input/Output. long WGS84CrsIntersect(LLPoint pt1, double crs13, double* crs31, double* dist13, LLPoint pt2, double crs23, double* crs32, double* dist23, LLPoint* intx, double tol) returns a reference to an LLPoint structure that contains the intersection coordinates, where the inputs are:

(1) LLPoint pt1 = Start point of first geodesic

(2) double crs13 = Azimuth from pt1 to intersection point

(3) double* crs31 = Reference to azimuth from intersection point to pt1

(4) double* dist13 = Reference to distance from pt1 to intersection

(5) LLPoint pt2 = Start point of second geodesic

(6) double crs23 = Azimuth from pt2 to intersection point

(7) double* crs32 = Reference to azimuth from intersection to pt2

(8) double* dist23 = Reference to distance between pt2 and intersection point

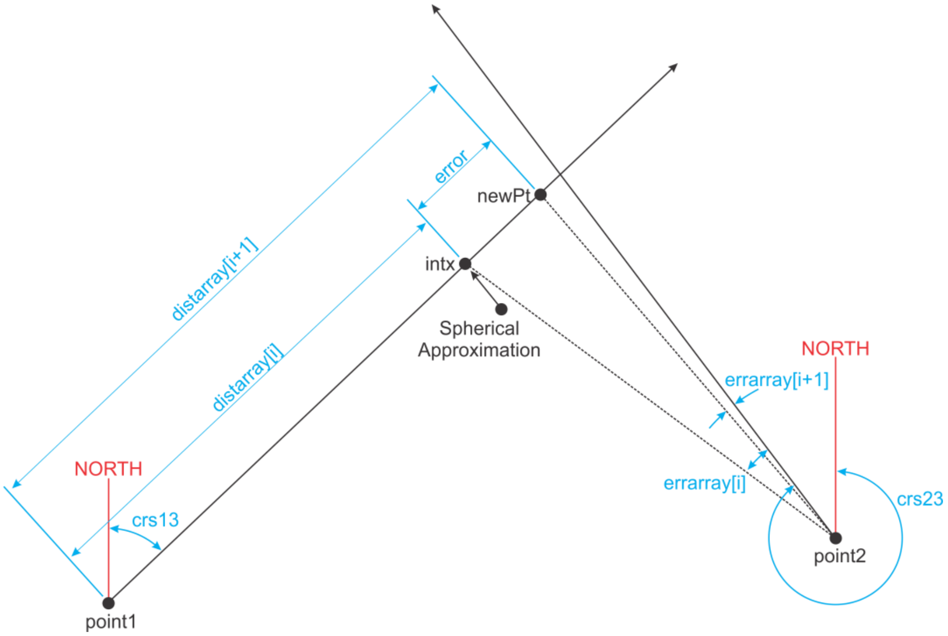
(9) LLPoint* intx = Reference to intersection point

(10) double tol = Maximum error allowed in solution

(11) double eps = Convergence parameter for forward/inverse algorithms

b. Algorithm steps.

Figure E-20. Finding the Intersection of Two Geodesics



(1) *Step 1*. Use inverse algorithm to calculate distance, azimuth and reverse azimuth from pt1 to pt2. Denote these values by dist12, crs21 and crs12, respectively. Run a check to see if pt1 lies on the geodesic defined by pt2 and crs23 and if pt2 lies on the geodesic defined by pt1 and crs13:

(a) If pt1 falls on geodesic 2 and pt2 falls on geodesic 1.

1. Return an error. Courses are collinear. There are infinite intersections.

2. If pt1 falls on geodesic 2. Return pt1.

3. If pt2 falls on geodesic 1. Return pt2.

(2) *Step 2*. Calculate the signed azimuth difference in angle between crs12 and crs13, denoted by angle1.

(3) *Step 3*. Calculate the signed azimuth difference in angle between crs21 and crs23, denoted by angle2.

(4) *Step 4*. If $(sin(angle1) \times sin(angle2) < 0)$ then the courses lay on opposite sides of the pt1-pt2 line and cannot intersect in this hemisphere. Use reciprocal course so that the nearest intersection may be found:

(a) If $(abs(angle1) > abs(angle2))$

1. $angle1 = (crs13 + \pi) - crs12$.

2. Else $angle2 = crs21 - (crs23 + \pi)$.

(5) *Step 5*. End if.

(6) *Step 6*. Locate the approximate intersection point, intx, using a spherical earth model [see section 3 paragraph 2].

(7) *Step 7*. The following steps describe the function iterateLineIntersection which is called once the initial approximation, intx, of the line intersection is found. The purpose of the iterateLineIntersection function is to further refine the solution.

(8) *Step 8*. Use the inverse algorithm to calculate dist13, the distance from pt1 to intx.

(9) *Step 9*. Use the inverse algorithm to calculate dist23, the distance from pt2 to intx.

(10) *Step 10*. If $(dist13 < tol)$ then the intersection point is very close to pt1.Calculation errors may lead to treating the point as if it were beyond the end of the geodesic. Therefore, it is helpful to move pt1 a small distance along the geodesic:

(a) Use the direct algorithm to move pt1 from its original coordinates 1 NM along azimuth $crs13 + \pi$.

(b) Use the inverse algorithm to calculate the azimuth acrs13 for the geodesic from the new pt1.

(11) *Step 11*. Repeat *Step*s *10*, *10(a)*, and *10(b)* for pt2 and crs23.

(12) *Step 12*. If $(dist23 < dist13)$ then the intersection point is closer to pt2 than pt1. In this case, the iterative scheme will be more accurate if we swap pt1 and pt2.This is because we iterate by projecting the approximate point onto the geodesic from pt1 and then calculating the error in azimuth from pt2. If the distance from pt2 to the intersection is small, then small errors in distance can correspond to large errors in azimuth, which will lead to slow convergence. Therefore, we swap the points so that we are always measuring azimuth errors farther from the geodesic starting point:

(a) $newPt = pt1$

(b) $pt1 = pt2$

(c) $pt2 = newPt$

(d) $acrs13 = crs13$

(e) $crs13 = crs23$

(f) $crs23 = acrs13$

(g) $dist13 = dist23$. We only need one distance so the other is not saved.

(h) $swapped = 1$. This is a flag that is set so that the solutions can be swapped back after they are found.

(13) *Step 13*. End if.

(14) *Step 14*. Initialize the distance array: $distarray[0] = dist13$. Errors in azimuth from pt2 will be measured as a function of distance from pt1. The two most recent distances from pt1 are stored in a two element array. This array is initialized with the distance from pt1 to intx.

(15) *Step 15*. Use the direct algorithm to project intx onto the geodesic from pt1. Use pt1 as the starting point, and a distance of distarray[0] and azimuth of crs13.

(16) *Step 16*. Use the inverse algorithm to measure the azimuth acrs23 from pt2 to intx.

(17) *Step 17*. Initialize the error array:

$errarray[0] = signedAzimuthDifference(acrs23, crs23)$

(18) *Step 18*. Initialize the second element of the distance array using a logical guess: $distarray[1] = 1.01 \times dist13$.

(19) *Step 19*. Use the direct algorithm to project the second approximation of intx onto the geodesic from pt1. Use pt1 as the starting point, and a distance of distarray[1] and azimuth of crs13.

(20) *Step 20*. Use the inverse algorithm to measure the azimuth acrs23 from pt2 to intx.

(21) *Step 21*. Initialize the error array:

$errarray[1] = signedAzimuthDifference(acrs23, crs23)$

(22) *Step 22*. Initialize $k = 0$

(23) *Step* 23. Do while $(k = 0)$ or $(error > tol)$ and $(k \leq MAX_{ITERATIONS})$:

(a) Use linear approximation to find root of errarray as a function of distarray. This gives an improved approximation to dist13.

(b) Use the direct algorithm to project the next approximation of the intersection point, newPt, onto the geodesic from pt1. Use pt1 as the starting point, and a distance of dist13 (calculated in previous step) and azimuth of crs13.

(c) Use inverse algorithm to calculate the azimuth acrs23 from pt2 to intx.

(d) Use the inverse algorithm to compute the distance from newPt to intx (the previous estimate). Denote this value as the error for this iteration.

(e) Update distarray and errarray with new values:

1. $distarray[0] = distarray[1]$

2. $distarray[1] = dist13$

3. $errarray[0] = errarray[1]$

4. $errarray[1] = signedAzimuthDifference(acrs23, crs23)$

(f) Increment k: $k = k + 1$

(24) *Step 24*. End while loop.

(25) *Step 25*. Check if k reached $\mathrm{MAX}_{\mathrm{ITERATIONS}}$. If so, then the algorithm may not have converged, so an error message should be displayed.

(26) *Step 26*. The distances and azimuths from pt1 and pt2 to intx are available at the end of this function, since they were calculated throughout the iteration. It may be beneficial to return them with the intx coordinates, since they may be needed by the calling function. If this is done, and if ($swapped = 1$) then the original identities of pt1 and pt2 were exchanged and the azimuths and distances must be swapped again before they are returned.

(27) *Step 27*. Return intx.

2. Intersection of two arcs. The following algorithm computes the intersection points of two arcs. Each arc is defined by its center point coordinates and radius. The algorithm will return a null solution (no points) if the arcs do not intersect; it will return a single set of coordinates if the arcs intersect tangentially; and it will return two sets of coordinates if the arcs overlap.

a. Input/Output. long WGS84ArcIntersect(LLPoint center1, double radius1, LLPoint center2, double radius2, LLPointPair intx, int* n, double tol) returns a reference to an LLPoint structure array that contains the coordinates of the intersection(s), where the inputs are:

(1) LLPoint center1 = Geodetic coordinates of first arc center

(2) double radius1 = Radius of first arc in nautical miles

(3) LLPoint center2 = Geodetic coordinates of second arc center

(4) double radius2 = Radius of second arc in nautical miles

(5) LLPointPair intx = Two-element array of LLPoint objects that will be updated with intersections' coordinates

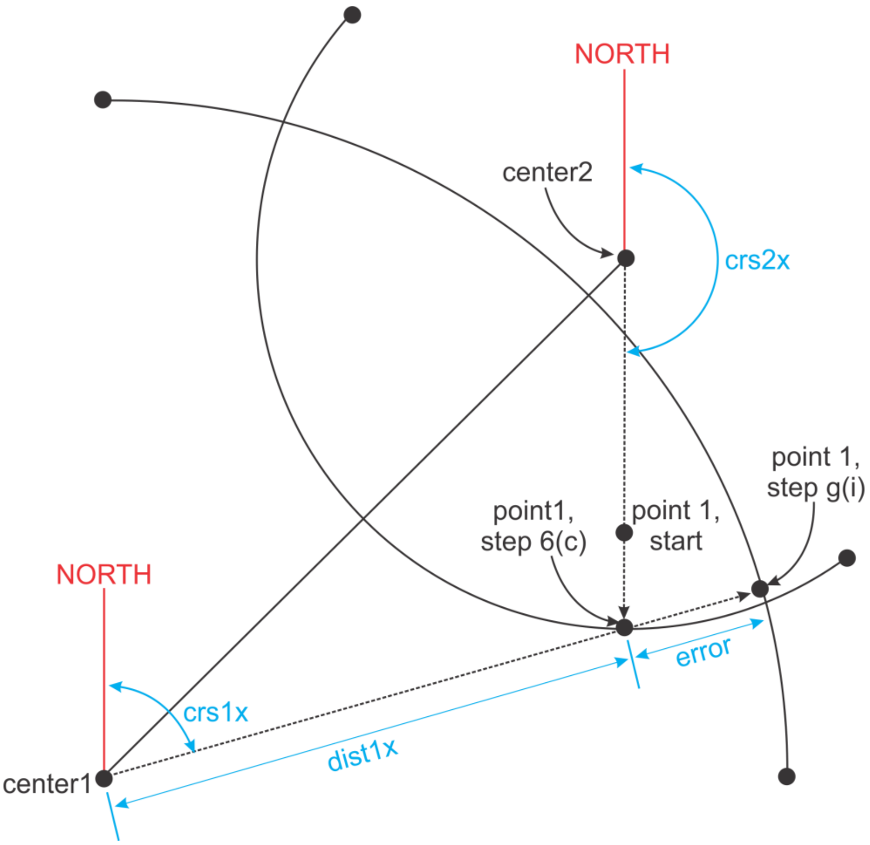
(6) int* n = Reference to integer number of intersection points returned

(7) double tol = Maximum error allowed in solution

(8) double eps = Convergence parameter for forward/inverse algorithms

b. Algorithm steps. This algorithm treats the arcs as full circles. Once the intersections of the circles are found, then each intersection point may be tested and discarded if it does not lie within the bounds of the arc.

Figure E-21. Intersection of Two Arcs



(1) *Step 1*. Use inverse algorithm to calculate the distance and azimuth between center1 and center2. Denote these values as dist12 and crs12, respectively.

(2) *Step 2*. If $(radius1 + radius2 - dist12 + tol < 0)$ or $(abs(radius1 - radius2) > dist12)$ then the circles are spaced such that they do not intersect. If the first conditional is true, then the arcs are too far apart. If the second conditional is true, then one arc is contained within the other. Return no intersections.

(3) *Step 3*. Else if $(abs(radius1 + radius2 - dist12) \leq tol)$ then the circles are tangent to each other and intersect in exactly one point:

(a) Use direct algorithm to project point from center1, along crs12, distance radius1.

(b) Return projected point.

(4) *Step 4*. End if.

(5) *Step 5*. Calculate approximate intersection points, intx[0] and intx[1], according to section 3 paragraph 2.

(6) *Step 6*. Iterate to improve approximation to pt:

(a) $k = 0$

(b) Use inverse algorithm to find azimuth from center2 to pt, denote this value as crs2x.

(c) Use direct algorithm to move pt along crs2x to circumference of circle 2. Use center2 as starting point, crs2x as azimuth, radius2 as distance.

(d) Use inverse algorithm to compute distance and azimuth from center1 to pt. Denote these values as dist1x and crs1x, respectively.

(e) Compute error at this iteration step: $error = radius1 - dist1x$.

(f) Initialize arrays to store error as function of course from center1:

1. $errarray[1] = error$

2. $crsarray[1] = crs1x$

(g) While ($k \leq maximumIterationCoun$t) and ($abs(errarray[1]) > tol$), improve approximation:

1. Use direct function to move pt along crs1x to circumference of circle1. Use center1 as starting point, crs1x as azimuth, and radius1 as distance. Note that crs1x was calculated as last step in previous iteration.

2. Use inverse function to find azimuth from center2 to pt, crs2x.

3. Use direct function to move pt along crs2x to circumference of circle2. Use center2 as starting point, crs2x as azimuth, and radius2 as distance.

4. Use inverse algorithm to compute distance and azimuth from center1 to pt. Denote these values as dist1x and crs1x, respectively.

5. Update function arrays:

a. $crsarray[0] = crsarray[1]$

b. $crsarray[1] = crs1x$

c. $errarray[0] = errarray[1]$

d. $errarray[1] = error$

6. Use linear root finder to find the azimuth value that corresponds to zero error. Update the variable crs1x with this root value.

7. Increment k: $k = k + 1$

(h) End while loop.

(7) *Step 7*. Store point in array to be returned: $intx[0] = point$.

(8) *Step 8*. Repeat *Step 6* for approximation intx[1].

(9) *Step 9*. Return array intx.

3. Intersections of arc and geodesic. The following algorithm computes the point where a geodesic intersects an arc. The geodesic is defined by its starting coordinates and azimuth. The arc is defined by its center point coordinates and radius. The algorithm will return a null solution (no points) if the arc and geodesic do not intersect; it will return a single set of coordinates if the arc and geodesic intersect tangentially; and it will return two sets of coordinates if the arc and geodesic overlap.

a. Input/Output. long WGS84LineArcIntersect(LLPoint pt1, double crs1, LLPoint center, double radius, LLPointPair intx, int* n, double tol) returns a reference to an LLPoint structure array that contains the coordinates of the intersection(s), where the inputs are:

(1) LLPoint pt1 = Geodetic coordinates of start point of geodesic

(2) double crs1 = Initial azimuth of geodesic at start point

(3) LLPoint center = Geodetic coordinates of arc center point

(4) double radius = Arc radius in nautical miles

(5) LLPointPair intx = Two-element array of LLPoint objects that will be updated with intersections' coordinates

(6) int* n = Reference to number of intersection points returned

(7) double tol = Maximum error allowed in solution

(8) double eps = Convergence parameter for forward/inverse algorithms

b. Algorithm steps. This algorithm treats the arc and geodesic as unbounded. Once intersection points are found, they must be tested using section 3 algorithms 5 and 6 to determine which, if any, lie within the curves’ bounds. This algorithm fails if the arc and geodesic describe the same great circle. A test for this case is embedded in *Step 7*.

Figure E-22. Locating First Intersection of Geodesic and Arc

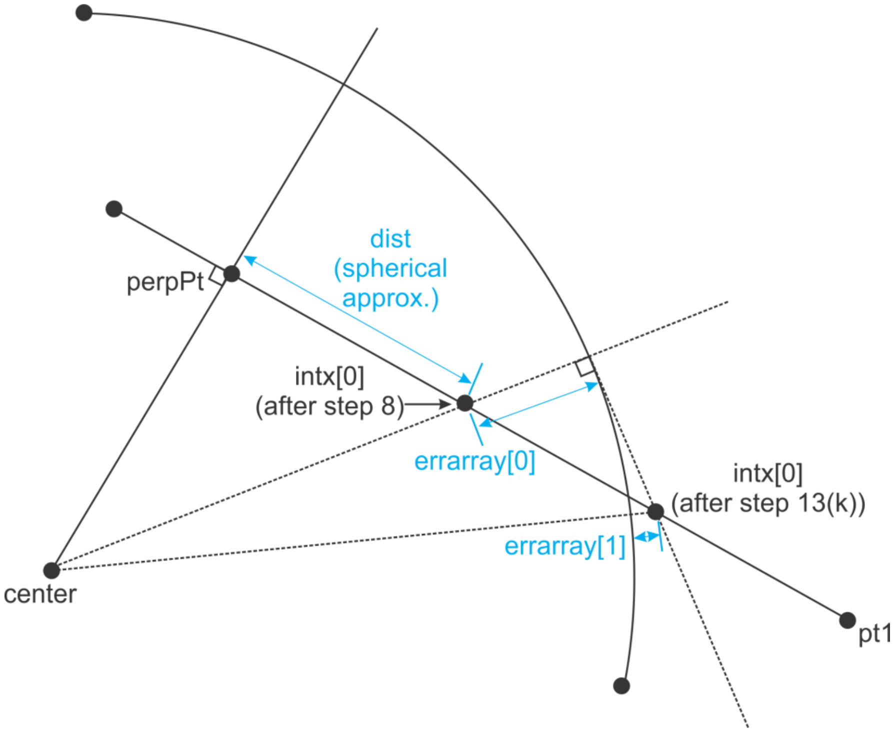
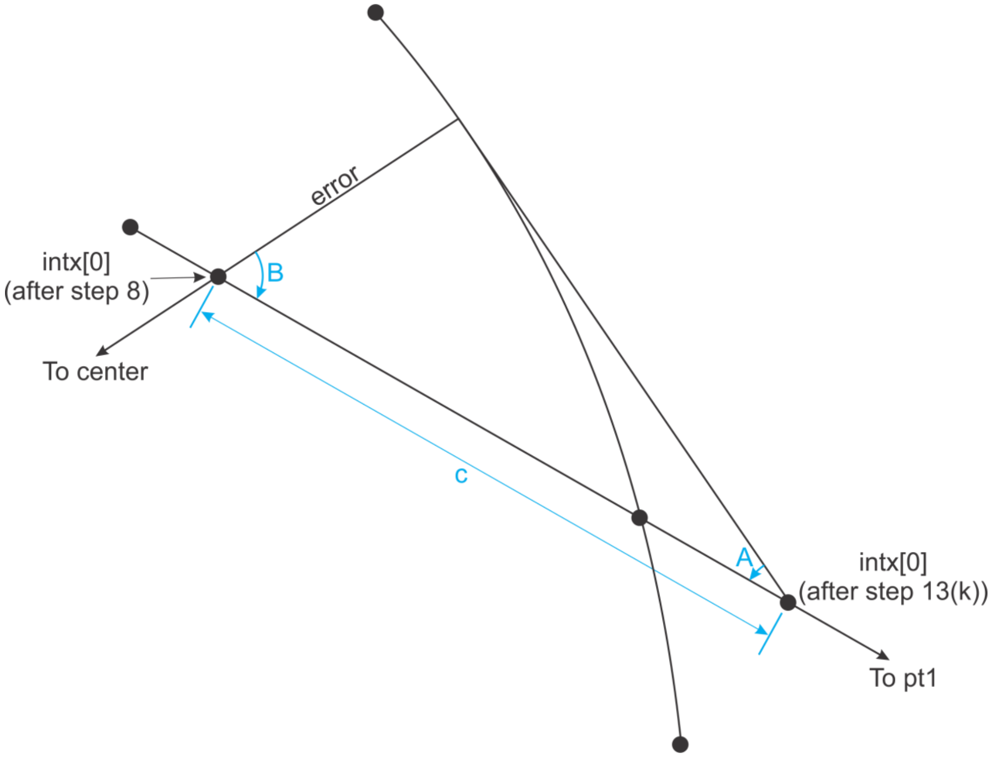


Figure E-23. Near the Appropriate Geodesic-Arc Intersection Point

With Spherical Triangle Components That Are Used to Improve the Solution



(1) *Step 1*. Use section 4 algorithm 1 to find the perpendicular projection point from arc center point (center) to the geodesic defined by starting point pt1 and azimuth crs1. Denote this point by perpPt. Denote the distance as perpDist.

(2) *Step 2*. Use inverse algorithm to calculate the azimuth of the geodesic at perpPt. Denote the azimuth from perpPt to pt1 as crs.

(3) *Step 3*. If $(abs(perpDist - radius) < tol)$ then the geodesic is tangent to the arc and intersection point is at perpPt, return $intx[0] = perpPt$.

(4) *Step 4*. Else if $(perpDist > radius)$ then geodesic passes too far from center of circle; there is no intersection, return empty array.

(5) *Step 5*. End if.

(6) *Step 6*. Use spherical triangle approximation to find distance from perpPt to one intersection points. Since the spherical triangle formed from center, perpPt, and either intersection point has a right angle at the perpPt vertex, the distance from perpPt to either intersection is:

$$dist = SPHERE_{RADIUS} \times \operatorname{acos}\left[\frac{\cos\left(\frac{radius}{SPHERE_{RADIUS}}\right)}{\cos\left(\frac{perpDist}{SPHERE_{RADIUS}}\right)}\right]$$

Note: A test must be performed so that if $(\cos\left(\frac{perpDist}{SPHERE_{RADIUS}}\right) = 0)$ then no solution is returned.

(7) *Step 7*. Find ellipsoidal approximation intx[0] to first intersection by starting at perpPt and using direct algorithm with distance dist and azimuth crs. This will place intx[0] on the geodesic.

(8) *Step 8*. Initialize iteration count $k = 0$.

(9) *Step 9*. Use inverse algorithm to calculate the distance from center to intx[0]. Denote this value by radDist. In the same calculation, calculate azimuth from intx[0] to center. Denote this value by rcrs; it will be used to improve the solution.

(10) *Step* 10. Calculate error for this iteration: $error = radius - \ radDist$.

(11) *Step 11*. Initialize arrays that will hold distance and error function values so that linear interpolation may be used to improve approximation:

(a) $distarray[0] = dist$

(b) $errarray[0] = error$

(12) *Step 12*. Do one iterative step using spherical approximation near intersection point:

(a) Use the inverse algorithm to calculate the azimuth from intx[0] to perpPt. Denote this value by bcrs.

(b) Compute the angle between the arc's radial line and the geodesic at intx[0].

$$B = abs(signedAzimuthDifference(bcrs, rcrs))$$

(c) Calculate the angle opposite the radial error:

$$A = \operatorname{acos}\left(\sin(B) \times \cos\left[\frac{abs(error)}{SPHERE_{RADIUS}}\right]\right)$$

(d) If $(abs(\sin(A)) < eps)$ then the triangle is nearly isosceles, so use simple formula for correction term c: $c = error$.

(e) Else if $(abs(A) < eps)$ then the error is very small, so use flat approximation:

$$c = \frac{error}{cos(B)}$$

(f) Else use a spherical triangle approximation for c:

$$c = SPHERE_{RADIUS} \times \operatorname{asin}\left[\frac{\sin\left(\frac{error}{SPHERE_{RADIUS}}\right)}{\sin(A)}\right]$$

(g) End if.

(h) If $(error > 0)$ then intx[0] is inside the circle, so approximation must be moved away from perpPt: $dist = dist + c$.

(i) Else $dist = dist - c$.

(j) End if.

(k) Use the direct algorithm to move intx[0] closer to solution. Use perpPt as the starting point with distance dist and azimuth crs.

(l) Use the inverse algorithm to calculate the distance from center to intx[0]. Denote this value again as radDist.

(m) Initialize second value of distarray and errarray:

1. $distarray[1] = dist$

2. $errarray[1] = radius - radDist$

(13) *Step 13*. Do while $(abs(error) > tol)$ and $(k < maximumIterationCount)$:

(a) Use a linear root finder to find the distance value that corresponds to zero error. Update the variable dist with this root value.

(b) Use the direct algorithm again to move intx[0] closer to solution. Use perpPt as the starting point with distance dist and azimuth crs.

(c) Use the inverse algorithm to calculate the distance from center to intx[0]. Denote this value radDist.

(d) Update distarray and errarray with the new values:

1. $distarray[0] = distarray[1]$

2. $errarray[0] = errarray[1]$

3. $distarray[1] = dist$

4. $errarray[1] = error$

(e) Increment the iteration count: $k = k + 1$

(14) *Step 14*. End while loop.

(15) *Step 15*. Prepare variables to solve for second solution, intx[1]:

(a) Second solution lies on other side of perpPt, so set $crs = crs + \pi$.

(b) Use direct algorithm to find intx[1]. Start at perpPt, using crs for the azimuth and dist for the distance, since the distance from perpPt to intx[0] is a very good approximation to the distance from perpPt to intx[1].

(c) Use inverse algorithm to calculate radDist, the distance from center to intx[1].

(d) Initialize the error function array: $errarray[0] = radius – radDist$.

(16) *Step 16*. Repeat *Step*s *13-14* to improve solution for intx[1].

(17) *Step 17*. Return intx[0] and intx[1].

4. Arc Tangent to two geodesics. This algorithm is useful for finding flight path arcs, such as fitting a fly-by turn or radius-to-fix (RF) leg between two track-to-fix (TF) legs. For the arc to be tangent to both the incoming and outgoing geodesics, the two tangent points must be different distances from the geodesics’ intersection point.

a. Input/Output. long WGS84TangentFixedRadiusArc(LLPoint pt1, double crs12, LLPoint pt3, double crs3, double radius, ArcDirection* dir, double tol) returns a reference to an LLPoint structure array that contains the coordinates of the center point and both tangent points of the arc that is tangent to both given geodesic, where the inputs are:

(1) LLPoint pt1 = Geodetic coordinates of start point of first geodesic

(2) double crs12 = Azimuth of first geodesic at pt1

(3) LLPoint pt3 = Geodetic coordinates of end point of second geodesic

(4) double crs3 = Azimuth of second geodesic at pt3

(5) double radius = Radius of desired arc

(6) ArcDirection* dir = Reference to an integer that represents direction of turn.

(a) dir = 1 for left hand turn

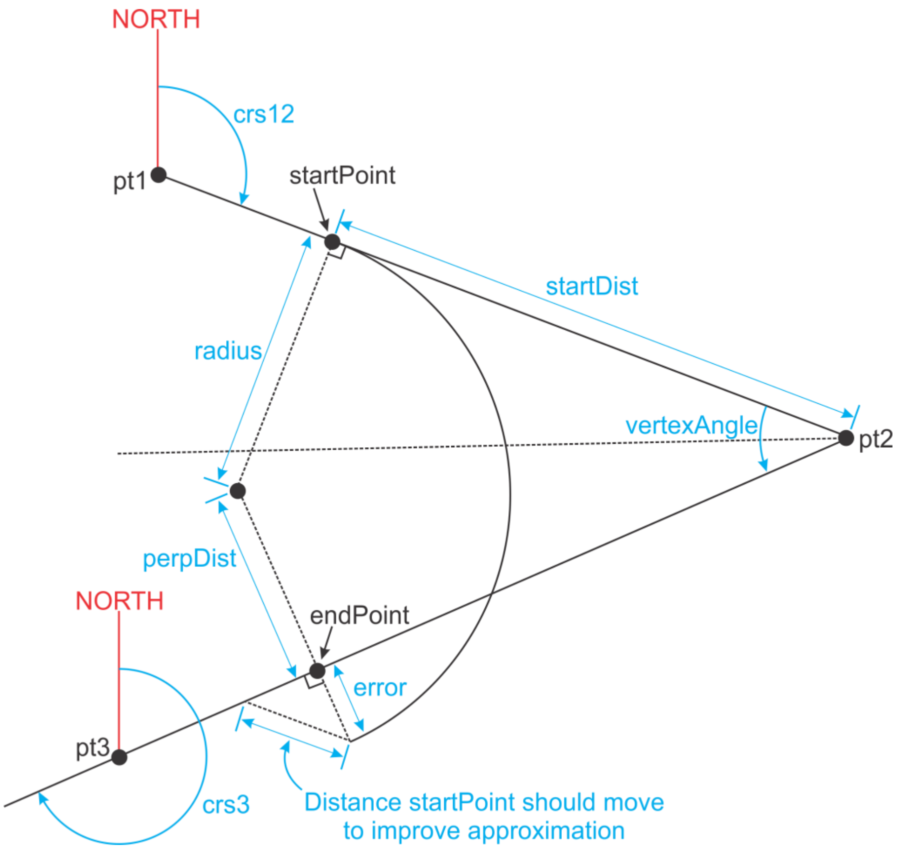
(b) dir = -1 for right hand turn

(7) double tol = Maximum error allowed in solution

(8) double eps = Convergence parameter for forward/inverse algorithms

b. Algorithm steps.

Figure E-24. Finding Arc Center and Points at Which Arc is Tangent to Two Geodesics



(1) *Step 1*. Use section 5 algorithm 1 to locate the intersection point of the given geodesics. The first geodesic has azimuth crs12 at pt1, while the second geodesic has azimuth crs3 at pt3. Denote their intersection point by pt2.

(2) *Step 2*. If intersection point pt2 is not found, then no tangent arc can be found, return empty array.

(3) *Step 3*. End if.

(4) *Step 4*. Use the inverse algorithm to calculate the distance from pt1 to pt2 (denoted by dist12). Also calculate the azimuth at pt2 to go from pt2 to pt1. Denote this value by crs21.

(5) *Step 5*. Use the inverse algorithm to compute the azimuth at pt2 to go from pt2 to pt3. Denote this value by crs23.

(6) *Step 6*. Calculate angle between courses at pt2 [see section 2 algorithm 3]. Denote this value by vertexAngle: $vertexAngle = signedAzimuthDifference(crs21, crs23)$

(7) *Step 7*. If $(abs(\sin(vertexAngle) < tol)$ then either there is no turn or the turn is 180 degrees. In either case, no tangent arc can be found, return empty array.

(8) *Step 8*. Else if $(vertexAngle > 0)$ then course changes direction to the right: $dir\ = -1$.

(9) *Step 9*. Else the course changes direction to the left: $dir = 1$

(10) *Step 10*. End if.

(11) *Step 11*. Use spherical triangle calculations to compute the approximate distance from pt2 to the points where the arc is tangent to either geodesic. Denote this distance by distToStart:

(a) $B = \frac{vertexAngle}{2}$

(b) If $(radius > SPHERE_{RADIUS} \times B)$ then no arc of the required radius will fit between the given geodesics, return empty array.

(c) End if.

(d) Calculate distToStart using the approximate formula from Napier's Rule of Circular Parts.

$$distToStart = SPHERE_{RADIUS} \times \text{asin}\left[\frac{\tan\left(\frac{radius}{SPHERE_{RADIUS}}\right)}{\tan(B)}\right]$$

(12) *Step 12*. Initialize the iteration count: $k = 0$

(13) *Step 13*. Initialize the error measure: $error = 0.0$

(14) *Step* 14. Do while $(k = 0)$ or $(abs(error) > tol)$ and $(k \leq maximumIterationCount)$

(a) Adjust the distance to tangent point based on current error value (this has no effect on first pass through, because $error = 0$):

$$distToStart = distToStart + \frac{error}{\sin(vertexAngle)}$$

(b) Use the direct algorithm to project startPt distance distToStart from pt1. Use pt1 as the starting point with azimuth of crs12 and distance of distToStart.

(c) Use the inverse algorithm to compute azimuth of geodesic at startPt. Denote this value by perpCrs.

(d) If (dir < 0) then the tangent arc must curve to the right. Add π/2 to perpCrs to get the azimuth from startPt to center of arc: $perpCrs = perpCrs + \frac{\pi}{2}$

(e) Else the tangent arc must curve to the left. Subtract π/2 from perpCrs to get the azimuth from startPt to center of arc: $perpCrs = perpCrs - \frac{\pi}{2}$

(f) End if.

(g) Use the direct algorithm to locate the arc center point, centerPoint. Use startPt as the starting point, perpCrs for the azimuth, and radius for the distance.

(h) Use section 4 algorithm 1 to project centerPoint to the second geodesic. Denote the projected point by endPt. This is approximately where the arc will be tangent to the second geodesic. Denote the distance from centerPoint to endPoint as perpDist.

(i) Calculate the tangency error: $error = radius - perpDist$. This error value will be compared against the required tolerance parameter. If its magnitude is greater than tol, then it will be used to adjust the position of startPoint until both startPoint and endPoint are the correct distance from centerPoint.

(15) *Step 15*. End while.

(16) *Step 16*. Return the values for centerPoint, the center of the arc, startPoint, the tangent point on the first geodesic, and endPoint, the tangent point of second geodesic.

5. Intersections of geodesic and locus. This algorithm is useful for finding the corner points of TF sub-segment's OEA, where a parallel (represented as a locus of points) intersects the geodesic end line.

a. Input/Output. long WGS84GeoLocusIntersect(LLPoint geoSt, LLPoint geoEnd, LLPoint* pint, Locus loc, double tol) returns a reference to an LLPoint structure array that contains the coordinates of the intersection point, where the inputs are:

(1) LLPoint geoSt = Geodetic coordinates of start point of geodesic

(2) LLPoint geoEnd = Geodetic coordinates of end point of geodesic

(3) Locus loc = Structure defining locus of points

(4) LLPoint* pint = Reference to LLPoint that will be updated with intersection coordinates.

(5) double tol = Maximum error allowed in solution

(6) double eps = Convergence parameter for forward/inverse algorithms

b. Algorithm steps.

Figure E-25. Intersection of Geodesic with Locus of Points

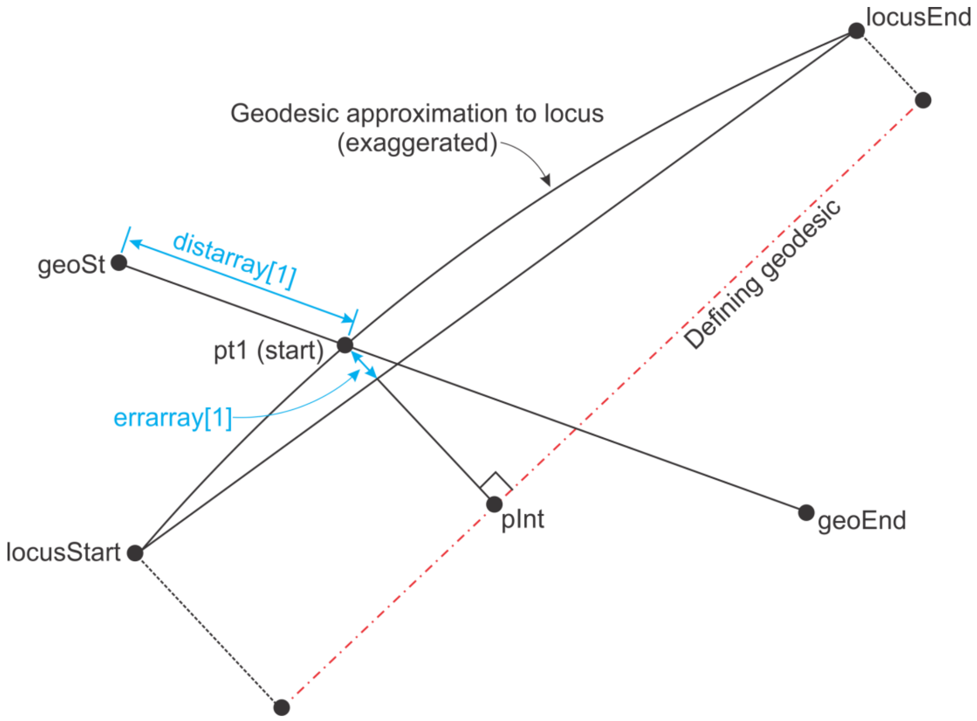
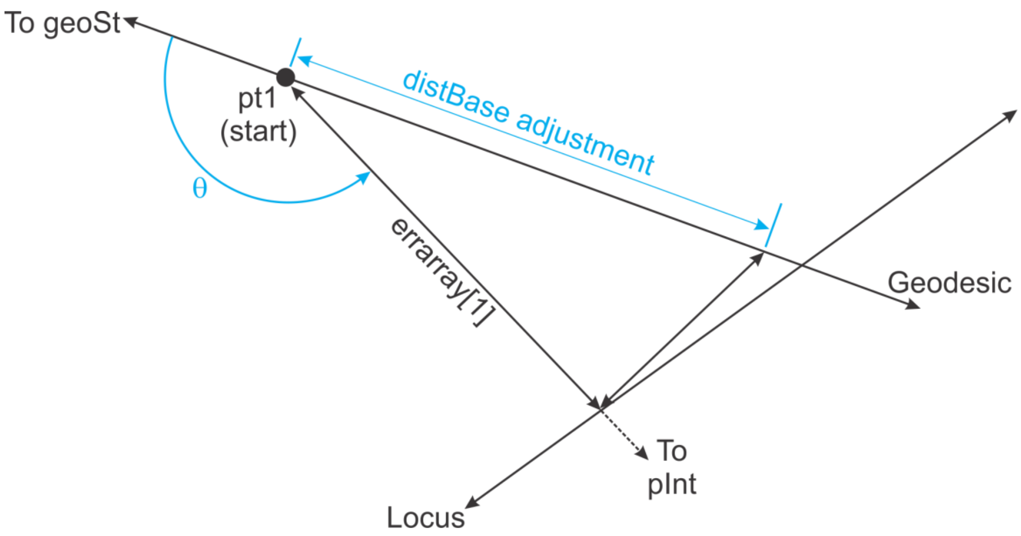


Figure E-26. Computing First Update to Locus-Geodesic Intersection



(1) *Step 1*. Use the geodesic intersection algorithm [see section 5 algorithm 1] to find a first approximation to the point where the given geodesic and locus intersect. Use the start and end coordinates of the locus along with the start and end coordinates of given geodesic as inputs to the geodesic intersection algorithm. This will erroneously treat the locus as a geodesic; however, the calculated intersection will be close to the desired intersection. The geodesic intersection algorithm will return the approximate intersection point, pt1, along with the courses and distances from the pt1 to the start points of the locus and given geodesic. Denote these courses and distances as crs31, dist13, crs32, dist23, respectively.

(2) *Step 2*. If pt1 is not found, then the locus and geodesic to not intersect, so return empty point.

(3) *Step 3*. End if.

(4) *Step 4*. Use the inverse algorithm to calculate the course from geoSt to geoEnd. Denote this value as fcrs. This value is needed by the direct algorithm to locate new points on the given geodesic.

(5) *Step 5*. Use the inverse algorithm to calculate the distance and course from pt1 to geoSt. Denote these values as distBase and crsBase, respectively.

(6) *Step 6*. Obtain the forward course of the locus's defining geodesic. This course is stored as loc.geoAz. Denote this course as tcrs. This value is needed to project the approximate point onto the defining geodesic in order to calculate the appropriate locus distance.

(7) *Step 7*. Use section 4 algorithm 1 to project pt1 onto the locus’s defining geodesic. Use pt1, loc.geoStart, and tcrs as inputs. Denote the returned point as pInt, the returned course as crsFromPt, and the returned distance as distFromPt.

(8) *Step 8*. Use section 3 algorithm 8 to calculate the distance from the defining geodesic to the locus at pInt. Denote this value as distLoc.

Note: distLoc may be positive or negative, depending on which side of defining geodesic the locus lays.

(9) *Step 9*. Calculate the distance from pt1 to the locus. This is the initial error:

$$errarray[1] = distFromPt - abs(distLoc)$$

(10) *Step 10*. Save the initial distance from geoSt to the approximate point: $geodarray[1] = distBase$. We will iterate to improve the approximation by finding a new value for distBase that makes errarray zero.

(11) *Step 11*. Calculate a new value of distBase that will move pt1 closer to the locus. This is done by approximating the region where the given geodesic and locus intersect as a right Euclidean triangle and estimating the distance from the current pt1 position to the locus.

(a) Calculate the angle between the geodesic from pt1 to pInt and the geodesic from pt1 to geoSt:

$$theta = abs(signedAzimuthDifference(crsFrompt, crsBase))$$

(b) Calculate a new value for distBase:

$$newdistbase = \frac{distbase - errarray[1]}{\cos(theta)}$$

(12) *Step 12*. Initialize the iteration count: $k = 0$.

(13) *Step 13*. Do while ($abs(errarray[1] > tol$) and ($k < maxIterationCount$):

(a) Use geoSt, fcrs, and newDistBase in the direct algorithm to update the value of pt1.

(b) Save the current values of errarray and geodarray:

<u>1.</u> $errarray[0] = errarray[1]$

<u>2.</u> $geodarray[0] = geodarray[1]$

(c) Set $geodarray[1] = newDistBase$.

(d) Repeat *Steps 7*, *8*, and *9* to calculate the distance from pt1 to the locus, distloc, and the corresponding update to errarray[1].

(e) Use a linear root finder with geodarray and errarray to find the distance value that makes the error zero. Update newDistBase with this root value.

(14) *Step 14*. End while.

(15) *Step* 15. Return $pint = pt1$.

6. Intersections of arc and locus. This algorithm solves for the intersection of a fixed radius arc and a locus. It is very similar to section 5 algorithm 3, which computes the intersections of an arc and a geodesic. It begins by treating the locus as a geodesic and applying section 5 algorithm 3 to find approximate intersection points. The approximation is improved by traveling along the locus, measuring the distance to the arc center at each point. The difference between this distance and the given arc radius is the error. The error is modeled as a series of linear functions of position on the locus. The root of each function gives the next approximation to the intersection. Iteration stops when the error is less than the specified tolerance.

a. Input/Output. long WGS84LocusArcIntersect(Locus loc, LLPoint center, double radius, LLPointPair intx, int* n, double tol) returns a reference to an LLPoint structure array that contains the coordinates of the intersection(s), where the inputs are:

(1) Locus loc = Locus of interest

(2) LLPoint center = Geodetic coordinates of arc

(3) double radius = Arc radius

(4) LLPointPair intx = Two-element array of LLPoint that will be updated with intersection coordinates

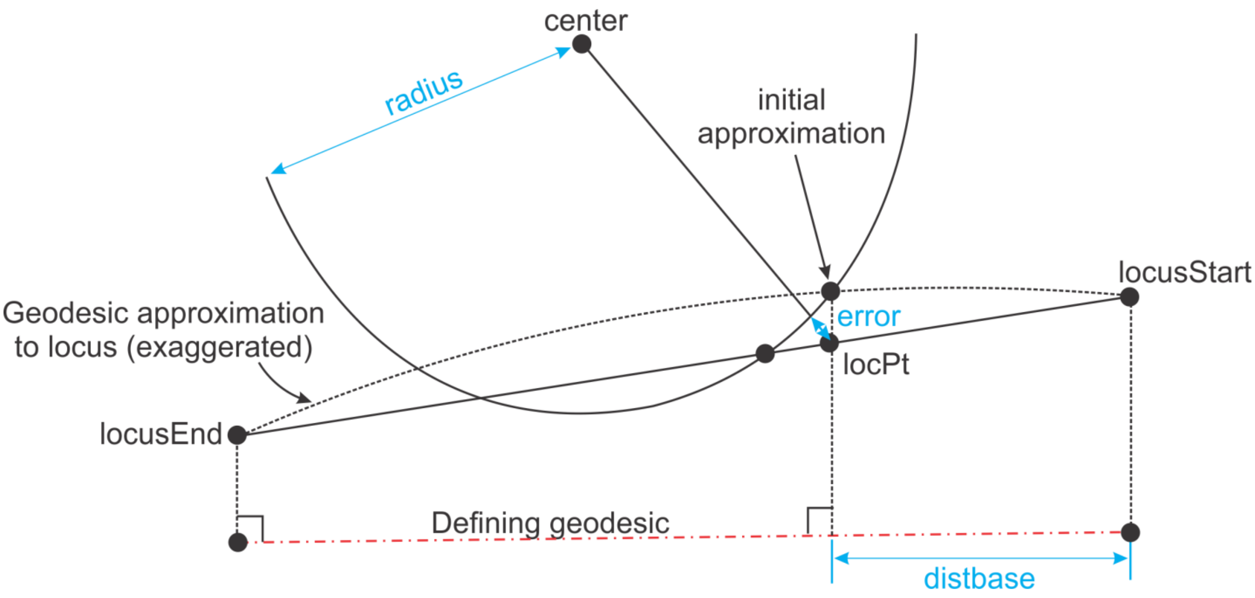
(5) int* n = Number of intersections found

(6) double tol = Maximum error allowed in solution

(7) double eps = Convergence parameter for forward/inverse algorithms

b. Algorithm steps.

Figure E-27. Finding the Intersection of an Arc and a Locus



(1) *Step 1*. Initialize number of intersections $n = 0$.

(2) *Step 2*. Use the inverse algorithm to compute the course from loc.locusStart to loc.locusEnd. Denote this value as fcrs.

(3) *Step 3*. Use section 4 algorithm 3 to project the center of the arc to the locus. Denote the projected point as locpt. Denote the distance and course from center to locpt as distFromPoint and crsFromPoint, respectively. If locpt is on or within the radius of the arc, then it will be used to find the intersection(s) of the locus and the arc, intx.

(4) *Step 4*. If $(distFromPoint > radius)$ then no approximate intersections were found. Return NULL.

(5) *Step 5*. End if.

(6) *Step 6*. Else if distFromPoint is equal to radius within tolerance level, then:

(a) Locus is tangent to arc. One intersection exists.

(b) $intx[0] = locpt$.

(7) *Step 7*. End if.

(8) *Step 8*. Otherwise, distFromPoint must be less than radius, meaning there are two possible intersections. These two approximate intersections are found using spherical trigonometry and the direct algorithm. Denote the approximate intersections as intx[0] and intx[1].

(9) *Step 9*. Use the inverse algorithm to compute the forward and reverse course from loc.geoStart to loc.geoEnd. Store these values as fcrs1 and bcrs, respectively.

(10) *Step 10*. For $i = 0, i < n1$:

(a) Use section 4 algorithm 1 to project intx[0] to the locus's defining geodesic. Denote the projected point as perpPt.

(b) Use the inverse algorithm to calculate distbase, the distance from perpPint to loc.geoStart.

(c) Use section 4 algorithm 2 to project locPt onto the locus from perpPint.

(d) Use the inverse algorithm to calculate distCent, the distance from locPt to center.

(e) Calculate the error and store it in an array:

$$errarray[1] = distCent - radius$$

(f) If $(abs(errarray[1]) < tol)$ then locPt is close enough to the circle. Set $intx[n] = locPt, n = n + 1$, and continue to the end of the "for" loop, skipping *Steps (10)(g)* through *(10)(m)*below.

(g) Save the current value of distbase to an array: $geodarray[1] = distbase.$

(h) Initialize the iteration count: $k = 0$

(i) Perturb distbase by a small amount to generate a second point at which to measure the error: $newDistbase = 1.001 \times distbase$

(j) Do while $(k < maxIterationCount)$ and $(abs(errarray[1]) < tol)$.

(k) Project Pt1 on the defining geodesic a distance newDistbase along course fcrs1 from loc.geoStart.

1. Use section 4 algorithm 2 to project locPt onto the locus from Pt1.

2. Use the inverse algorithm to calculate dist1, the distance from locPt to center.

3. Calculate the error: $error = dist1 - radius.$

4. Update the distance and error arrays:

a. geodarray[0] = geodarray[1]

b. geodarray[1] = newDistbase

c. errarray[0] = errarray[1]

d. errarray[1] = error

5. Use a linear root finder with geodarray and errarray to find the distance value that makes the error zero. Update newDistbase with this root value.

(l) End while.

(m) If locPt is on the locus according to section 3 algorithm 9, then

1. Copy locPt to the output array: $intx[n] = locPt$.

2. Update the count of intersection points found: $n = n + 1$.

(11) *Step 11*. End for loop.

(12) *Step 12*. Return intx.

7. Intersections of two loci.

a. Input/Output. long WGS84LocusIntersect(Locus loc1, Locus loc2, LLPoint* intx, double tol) returns a reference to an LLPoint structure array that contains the intersection coordinates, where the inputs are:

(1) Locus loc1 = First locus of interest

(2) Locus loc2 = Second locus of interest

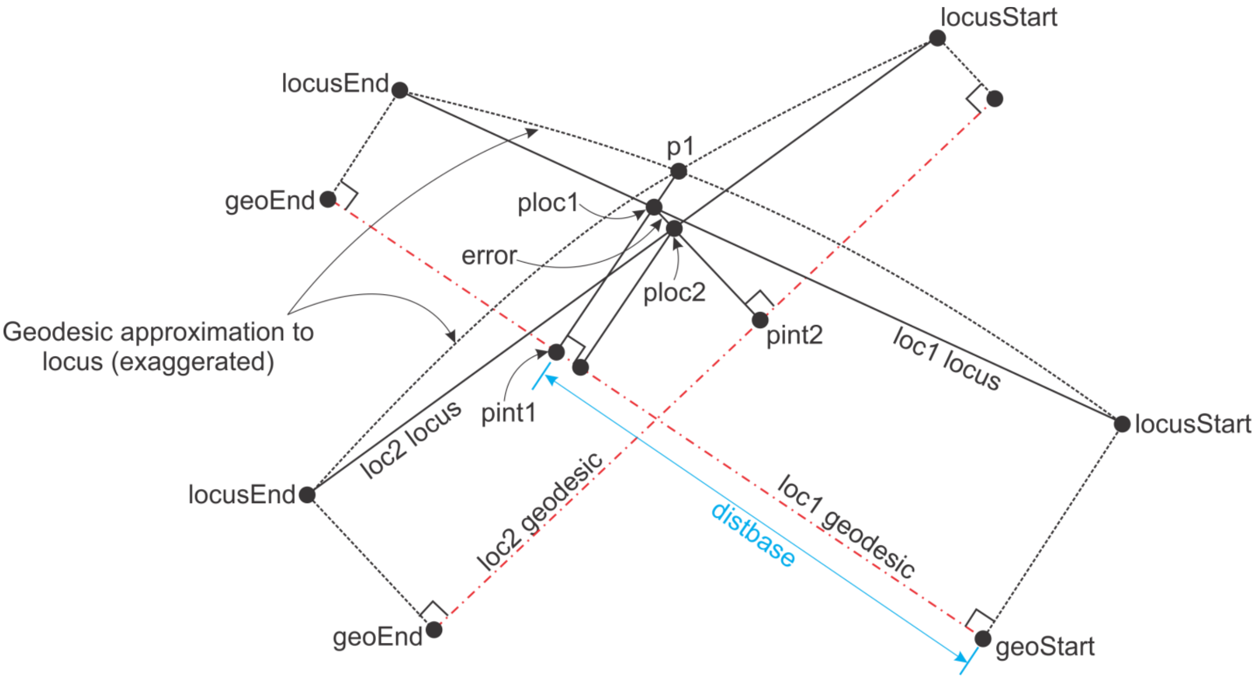
(3) LLPoint* intx = Reference to LLPoint that will be updated with intersection coordinates.

(4) Double tol = Maximum error allowed in solution

(5) Double eps = Convergence parameter for forward/inverse algorithms

b. Algorithm steps.

Figure E-28. Computing the Intersection of Two Loci



(1) *Step 1*. Use the inverse algorithm to calculate the course of the geodesic approximation to loc1. Use loc1.locusStart and loc1.locusEnd as start and end points. Denote this course as crs1.

(2) *Step 2*. Use the inverse algorithm to calculate the course of the geodesic approximation to loc2. Use loc2.locusStart and loc2.locusEnd as start and end points. Denote this course as crs2.

(3) *Step 3*. Use loc1.locusStart, crs1, loc2.locusStart, and crs2 as input to section 5 algorithm 1 to calculate an approximate solution to the locus intersection. Denote the approximate intersection point at p1.

(4) *Step 4*. If $(p1 = NULL)$ then the loci do not intersect, so return NULL.

(5) *Step 5*. Use the inverse algorithm to calculate the course of loc1's defining geodesic. Use loc1.geoStart and loc1.geoEnd as the start and end points, and denote the course as tcrs1.

(6) *Step 6*. Project p1 to the geodesic of loc1 using section 4 algorithm 1 with loc1.geoStart and tcrs1 as input parameters. Store the projected point as pint1.

(7) *Step 7*. If $(pint1 = NULL)$ then no projected point was found so return NULL.

(8) *Step 8*. Use the inverse algorithm to calculate distbase, the distance from loc1.geoStart to pint1.

(9) *Step 9*. Initialize iteration counter: $k = 0$

(10) *Step 10*. Do while $(k = 0)$ or $(k < maxIterationCount)$ and $(abs(error) > tol)$

(a) If $(k > 0)$ then apply direct algorithm to project new pint1 on loc1. Use starting point loc1.geoStart, course tcrs1, and distance distbase.

(b) Use section 4 algorithm 2 to project a point on loc1 from the current pint1. Denote the projected point as ploc1.

(c) Project ploc1 to the geodesic of loc2 using section 4 algorithm 1 with loc2.geoStart and tcrs2 as input parameters. Store the projected point as pint2.

(d) Use section 4 algorithm 2 to project a point on loc2 from pint2. Denote the projected point as ploc2. If ploc1 were truly at the intersection of the loci, then ploc2 and ploc1 would be the same point. The distance between them measures the error at this calculation step.

(e) Compute the error by using the inverse algorithm to calculate the distance between ploc1 and ploc2.

(f) Update the error and distance arrays and store the current values:

1. $errarray[0] = errarray[1]$

2. $errarray[1] = error$

3. $distarray[0] = distarray[1]$

4. $distarray[1] = distbase$

(g) If $(k = 0)$ then project ploc2 onto loc1 to get a new estimate of distbase:

1. Project ploc2 to the geodesic of loc1 using section 4 algorithm 1 with loc1.geoStart and tcrs1 as input parameters. Store the projected point as pint1.

2. Use the inverse algorithm to calculate distbase, the distance from loc1.geoStart to pint1.

(h) Else use a linear root finder with distarray and errarray to find the distance value that makes the error zero. Update distbase with this root value. This is possible only after the first update step because two values are required in each array.

(i) End if.

(j) Increment iteration count: $k = k + 1$.

(11) *Step 11*. End while.

(12) *Step 12*. Use section 3 algorithm 9 with inputs of loc1 and ploc1 to determine if ploc1 lies on the loc1. Then use section 3 algorithm 9 with inputs of loc2 and ploc1 to determine if ploc1 lies on the loc2. If ploc1 does not lie on both loci, return NULL.

(13) *Step 13*. Return ploc1.

8. Arc tangent to two loci. Computing a tangent arc of a given radius to two loci is very similar to fitting an arc to two geodesics. The following algorithm uses the same basic logic as section 5 algorithm 4.

a. Input/Output. long WGS84LocusTanFixedRadiusArc(Locus loc1, Locus loc2, double radius, LLPoint* centerPoint, LLPoint* startPoint, LLPoint* endPoint, ArcDirection* dir, double tol) returns a reference to an LLPoint structure array that contains the coordinates of the center point and both tangent points of the arc that is tangent to both given loci, where the inputs are:

(1) Locus loc1 = Structure defining first locus

(2) Locus loc2 = Structure defining second locus

(3) double radius = Radius of desired arc

(4) LLPoint* centerpoint = Reference to LLPoint that will contain arc's center coordinates.

(5) LLPoint* startPoint = Reference to LLPoint that will contain arc's start point coordinates.

(6) LLPoint* endpoint = Reference to LLPoint that will contain arc's endpoint coordinates.

(7) ArcDirection* dir = Reference to an integer that represents direction of turn.

(a) dir = 1 for left hand turn

(b) dir = -1 for right hand turn

(8) double tol = Maximum error allowed in solution

(9) double eps = Convergence parameter for forward/inverse algorithms

b. Algorithm steps.

Figure E-29. Arc Tangent to Two Loci

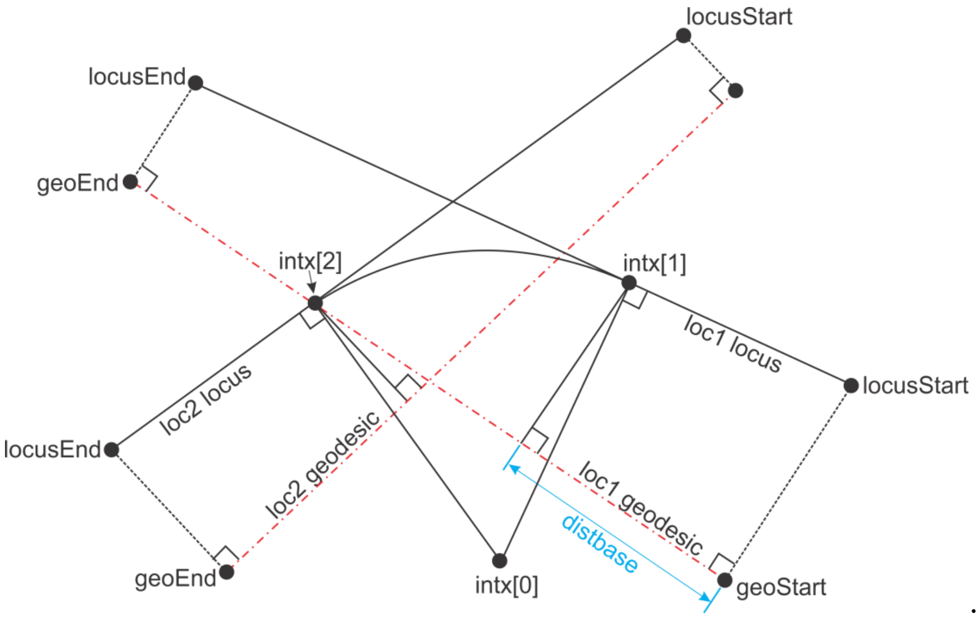
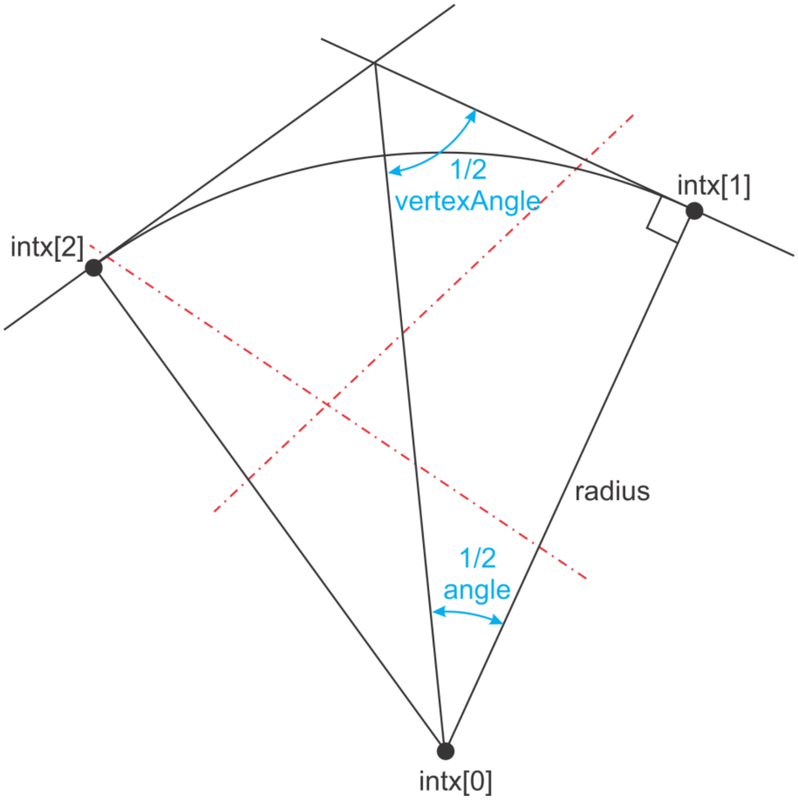


Figure E-30. Spherical Triangle Construction Used for Calculating the Approximate Vertex Angle at the Intersection of Two Loci



(1) *Step 1*. Use inverse algorithm to calculate crs12, the course from loc1.locusStart to loc1.locusEnd.

(2) *Step 2*. Use inverse algorithm to calculate gcrs1 and geoLen1, the course and distance from loc1.geoStart to loc1.geoEnd.

(3) *Step 3*. Use inverse algorithm to calculate crs32, the course from loc2.locusEnd to loc2.locusStart. Convert crs32 to its reciprocal: $crs32 = crs32 + \pi$.

(4) *Step 4*. Apply section 5 algorithm 4 to find the arc tangent to the geodesic approximations to loc1 and loc2. Use loc1.locusStart, crs12, loc2.locusEnd, crs32, and radius as input parameter. Denote the array of points returned as intx. intx[0] will be the approximate arc center point, intx[1] will be the tangent point near loc1, and intx[2] will be the tangent point near loc2. Also returned will be the direction of the arc, dir.

(5) *Step 5*. If $(intx = NULL)$ then there is no tangent arc. Return NULL.

(6) *Step 6*. Calculate the approximate angle at the vertex where loc1 and loc2 intersect. This will be used only to estimate the first improvement to the tangent point intx[1]. Thus we use an efficient spherical triangles approximation:

(a) Use the spherical inverse function to calculate the rcrs1, the course from intx[0] (the approximate arc center) to intx[1] (the approximate tangent point on loc1).

(b) Use the spherical inverse function to calculate the rcrs2, the course from intx[0] to intx[2] (the other approximate tangent point).

(c) Calculate the angle difference between rcrs1 and rcrs2:

$$angle = abs(signedAzimuthDifference(rcrs1, rcrs2))$$

$$vertexAngle = 2 \times \operatorname{acos}\left[\sin\left(\frac{angle}{2}\right) \times \cos\left(\frac{radius}{SPHERE_{RADIUS}}\right)\right]$$

(7) *Step 7*. Calculate the inclination angle of loc1 relative to its geodesic:

$$locAngle = \operatorname{atan}\left(\frac{loc1.endDist - loc1.startDist}{geoLen1}\right)$$

(8) *Step* 8. Initialize $distbase = 0.1$.

(9) *Step 9*. Initialize the iteration count: $k = 0$.

(10) *Step 10*. Do while $(k = 0)$ or $(k < maxiterationCount)$ and $(abs(error) > tol)$

(a) Use direct algorithm with starting point loc1.geoStart, course gcrs1, and distance distbase to project point geoPt.

(b) Use section 4 algorithm 2 to project a point on loc1 from the current geoPt1. Denote the projected point as intx[1].

(c) Use section 3 algorithm 10 to calculate lcrs1, the course of loc1 at intx[1].

(d) Convert lcrs1 into the correct perpendicular course toward the arc center (note that $dir > 0$ indicates a left-hand turn): $lcrs1 = lcrs1 - dir \times \frac{\pi}{2}$.

(e) Use the direct algorithm with starting point intx[1], course lcrs1, and distance radius to project the arc center point, intx[0].

(f) Use section 4 algorithm 3 to project intx[0] onto loc2. Reassign intx[2] as the projected point.

(g) Use the inverse algorithm to calculate r2, the distance from intx[0] to intx[2].

(h) Calculate the error: $error = r2 - radius$.

(i) Update the distance and error function arrays:

1. $distarray[0] = distarray[1]$

2. $distarray[1] = distbase$

3. $errarray[0] = errarray[1]$

4. $errarray[1] = error$

(j) If ($k = 0$) then estimate better distbase value using spherical approximation and calculated error:

$$distbase = distbase + error \times \frac{\cos(locAngle)}{\sin(vertexAngle)}$$

(k) Else use a linear root finder with distarray and errarray to find the distance value that makes the error zero. Update distbase with this root value.

(l) End if.

(11) *Step 11*. End while.

(12) *Step 12*. Return intx.