Vertical Aircraft Navigation

Rony Lanssiers

Rev 0 - February 23, 2019
Rev 1 - April 16, 2019

Abstract

After a short introduction into vertical navigation related terminology, the basic formulae are derived allowing a pilot of a fixed-wing aircraft to calculate the lateral distance needed to make a given descent and to calculate the rate at which the airplane has to descend to remain on a chosen glide slope. The reader remains responsible for verifying the correctness of the formulae and graphs in this document and their correct application.

1 Introduction

Vertical navigation refers to an aircraft's movements in vertical direction and is particularly important during the climb and descent phases of a flight. The full flight trajectory is determined by the combination of vertical and lateral navigation, the latter being an aircraft's movements in the horizontal plane.

This document focuses on the vertical flight profile when approaching the destination airport. It is of utmost importance for a pilot to know at what distance (s)he has to initiate the descent and at what rate.

Not only does the airplane have to be at the right altitude at the right place to touch down at the start of the runway, there are often altitude constraints to be met at different waypoints along the approach route.

2 Vertical distances

In aviation, different terminology is used to denote vertical distances depending on the reference plane which is used to measure the distance from:

Height is an aircraft's vertical distance above ground level and is usually expressed in feet above ground level, abbreviated as "ft AGL". The correct reference plane is entered in the aircraft's altimeters by setting their barometric reference pressure to the local barometric pressure at the airfield, a value commonly known as QFE. On the ground, the aircraft's altimeters consequently indicate zero.

Altitude is an aircraft's vertical distance above mean sea level and is usually expressed in feet above mean sea level, abbreviated as "ft AMSL". The correct reference plane is entered in the aircraft's altimeters by setting their barometric reference pressure to the local barometric pressure at the airfield reduced to mean sea level, a value commonly known as QNH. On the ground, the aircraft's altimeters consequently indicate the airfield's elevation above mean sea level.

Flight level is an aircraft's vertical distance above the barometric reference plane of $1013.25\ \mathrm{hPa}$ or $29.92\ \mathrm{in\ Hg}$ and is expressed in hundreds of feet or flight levels (e.g. FL240). The correct reference plane is entered in the aircraft's altimeters by setting their barometric reference pressure to $1013.25\ \mathrm{hPa}$ or $29.92\ \mathrm{in\ Hg}$, a value commonly known as standard pressure. On the ground, the aircraft's altimeters consequently indicate a value which depends on the difference between standard pressure and the true barometric pressure at the airfield.

The relationship between height, altitude and flight level is illustrated in figure 1.

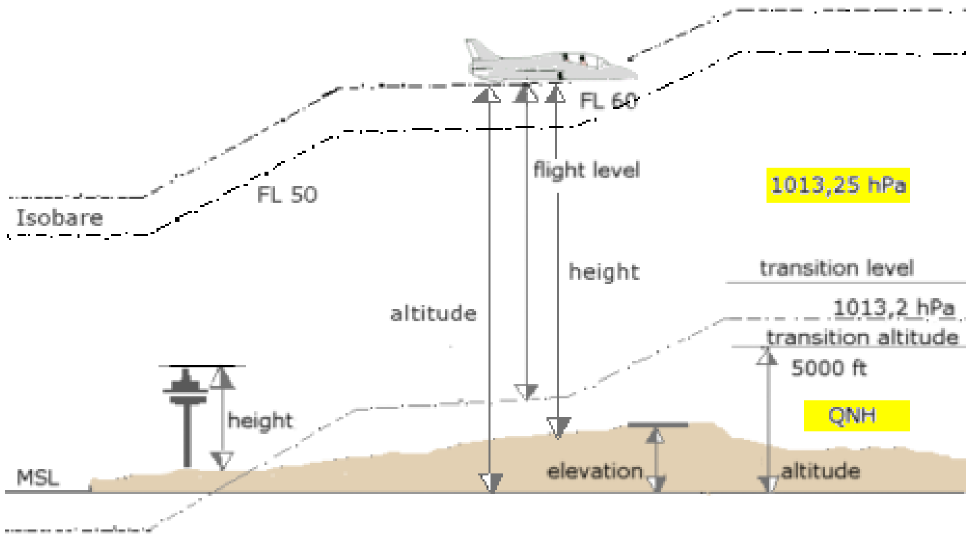


Figure 1: Relationship between height, altitude and flight level. (Credit: Wessmann @ Wikimedia Commons)

It would be very inconvenient if the altimeter setting would regularly have to be adjusted to the local barometric pressure reduced to sea level while the flight progresses. Airliners therefore transition from altitudes to flight levels as soon as there is no longer a risk of colliding with ground obstacles. In cruise flight, the reference plane is then the same for all aircraft and does not change, no matter how long the flight takes. This makes it much more easier to maintain a safe vertical separation between aircraft. The transition altitude at which the altimeters have to be set to standard pressure in indicated on air navigation charts as shown in figure 2. During descent, the transition level and QNH to be used for the approach are usually communicated to the cockpit crew by Air Traffic Control.

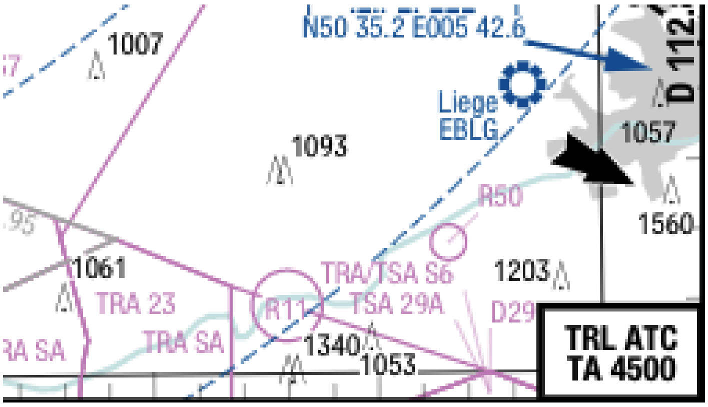


Figure 2: Transition altitude indicated in the bottom right corner of an air navigation chart. (Credit: Lufthansa Systems)

3 Lateral distance required to make a descent

The descent profile of an aircraft is schematized in figure 3.

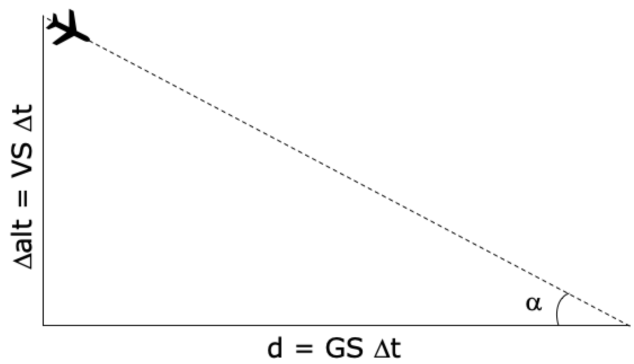


Figure 3: Schematized descent profile in which Δalt is the vertical distance to loose, VS the vertical speed, Δt an arbitrary time interval, d the lateral distance required to make the descent, GS the ground speed and α the glide slope angle.

Basic trigonometry yields:

$$\tan(\alpha) = \frac{\Delta alt}{f_1 \; d} \tag{1}$$

Factor f_1 in equation (1) is nothing more than a unit conversion factor and equals approximately 6076 ft/Nm as vertical distances are usually expressed in feet ($1 \text{ ft} = 0.3048 \text{ m}$) while lateral distances are expressed in nautical miles ($1 \text{ Nm} = 1852 \text{ m}$).

Using equation (1), the lateral distance d required to make the descent can be expressed as a function of the vertical distance Δalt to loose, for any given desired glide slope angle α:

$$\boxed{d \approx \frac{\Delta alt}{6076 \; \tan(\alpha)}} \tag{2}$$

Equation (2) is plotted in figure 4 with d as x-axis and Δalt as y-axis for α ranging from 2° to 5°. By entering the graph via the y-axis with the vertical distance to loose and finding the intersection with the line of the desired glide slope angle, the corresponding lateral distance required to make the descent can be found on the x-axis. The MATLAB [1] source code which was used to create figure 4 is given in appendix A.

[1] https://www.mathworks.com

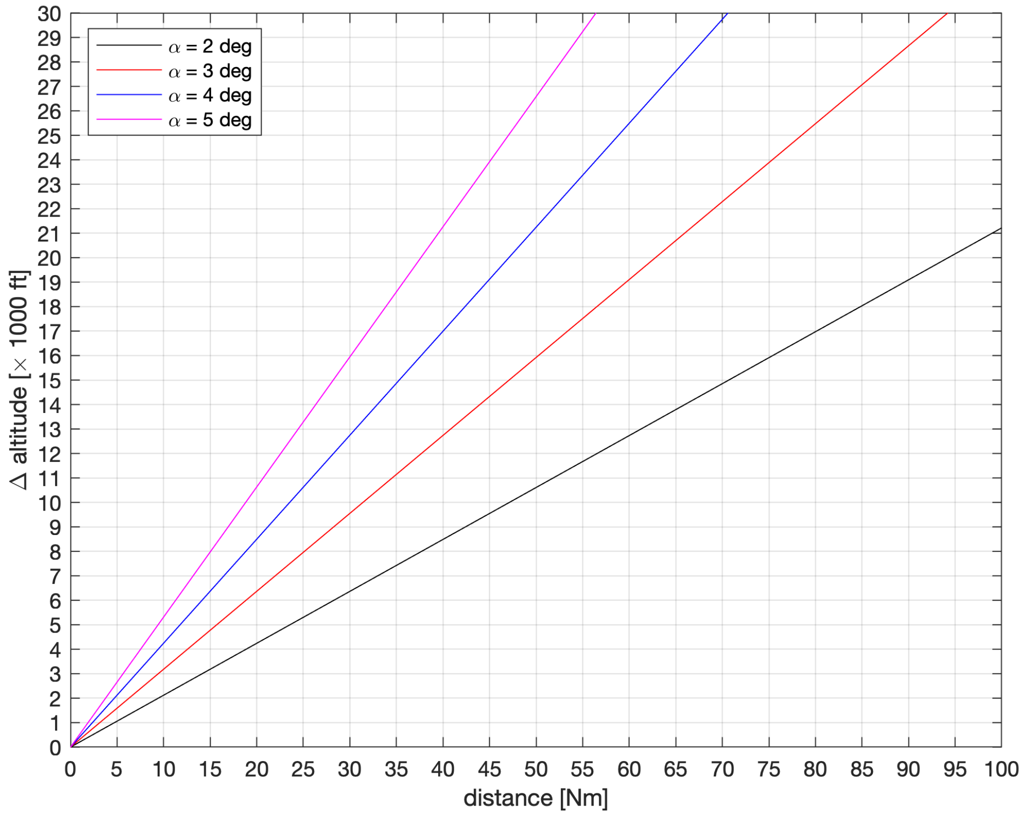


Figure 4: Vertical distance to loose versus lateral distance required to make the descent for different glide slope angles.

The glide slope angle which has to be maintained during final approach is indicated on approach charts as illustrated in figure 5. An angle of $3°$ is very common but there are airports like London City Airport where a steeper descent is required.

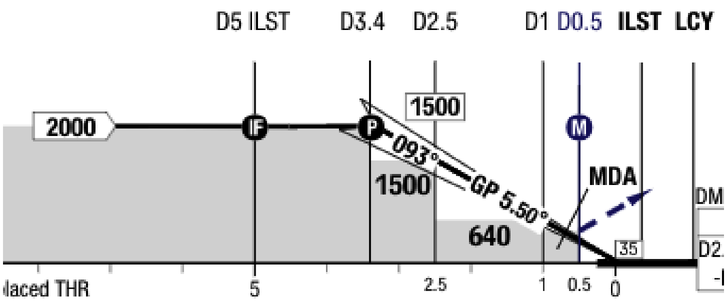


Figure 5: Part of an approach chart indicating the required glide path of $5.50°$ during final approach to London City Airport runway 09. (Credit: Lufthansa Systems)

4 Required rate of descent

Once the lateral distance required to make the descent has been determined, the next question which arises is at what rate the aircraft has to descend to remain on the desired glide slope. The governing equation can be derived by substituting the distances in equation (1) with their equivalents based on the corresponding speeds. Indeed, if VS is the aircraft's vertical speed and GS its (horizontal) ground speed, then for any time interval Δt:

$$\Delta alt = VS\ \Delta t$$

$$d = GS\ \Delta t$$

Equation (1) then becomes:

$$\tan(\alpha) = \frac{f_2\ VS\ \Delta t}{f_1\ GS\ \Delta t}$$

Factor f_2 is again a unit conversion factor and equals $60\ \mathrm{min/h}$ as vertical speeds are usually expressed as a distance per minute while ground speeds are expressed as a distance per hour.

Rearranging factors yields:

$$\boxed{VS \approx 101.3\ \tan(\alpha)\ GS} \tag{3}$$

The sign of VS is of no concern in this document. The same formulae can therefore be applied for climb and descent profiles, implicitly taking into account that the vertical speed on descent is obviously negative. For a rigorous mathematical treaty, descent glide slope angles can be given a negative sign.

Equation (3) is plotted in figure 6 with GS as x-axis and VS as y-axis for α ranging from $2°$ to $5°$. By entering the graph via the x-axis with the ground speed and finding the intersection with the line of the desired glide slope angle, the corresponding vertical speed can be found on the y-axis. The MATLAB source code which was used to create figure 6 is given in the appendices.

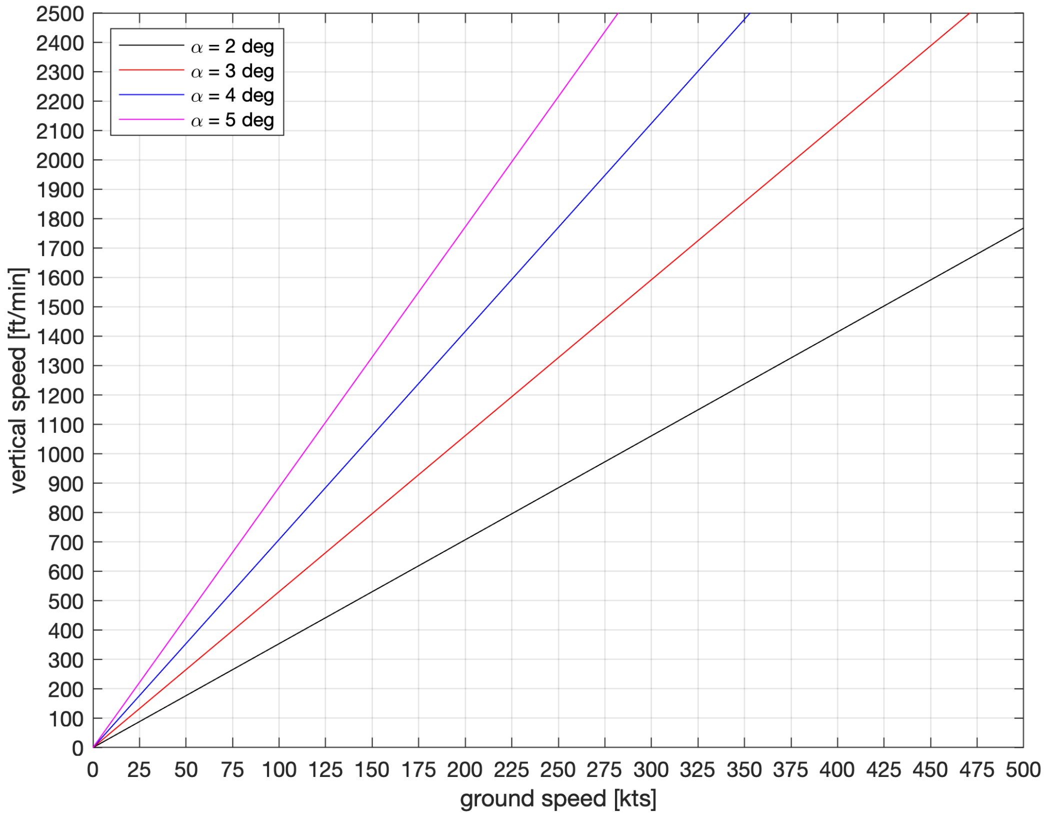


Figure 6: Vertical speed versus ground speed for different glide slope angles.

5 Rules of thumb

The approximate proportionality factor between the lateral distance required to make the descent and the vertical distance to loose and the approximate proportionality factor between the vertical speed and the ground speed for different glide slope angles are listed in table 1. They can be used as rules of thumb instead of the graphs.

α [°]	$d/\Delta alt$ [Nm/1000 ft]	VS/GS [ft/min/kt]
2	4.7	4
3	3.1	5
4	2.4	7
5	1.9	9

Table 1: Approximations for lateral distance required per 1000 feet descent and vertical speed per knot ground speed for different glide slope angles.

As an example, consider an aircraft flying at a ground speed of 250 kts which needs to loose a vertical distance of 8000 ft. From table 1, it can be derived that on a glide slope of $3°$, the descent will require a lateral distance of $8 \times 3.1 = 24.8$ Nm and a vertical speed of $250 \times 5 = 1250$ ft/min.

6 Simulator demo

The lua[2] script given in appendix B makes it possible to visualize equation (3) in the desktop flight simulator X-Plane 11[3]. The script makes use of the functionality of the FlyWithLua[4] simulator plugin to draw a gauge on the left side of the screen, indicating the aircraft's current vertical speed in relation to the vertical speed required for a glide slope angle between $2°$ and $5°$. Figure 7 is a detail of the left hand side of the opening figure and illustrates the case of a Boeing 737-800 on a $3°$ ILS approach. The gauge can be toggled on/off by binding the custom command "RLA/VNAV/ToggleGauge" to a joystick or keyboard button.

Figure 7: Detail of the opening figure, showing the actual vertical speed in relation to the vertical speed required for a glide slope angle between $2°$ and $5°$.

[2]Lua is a lightweight and embeddable scripting language designed, implemented and maintained by a team at the Pontifical Catholic University of Rio de Janeiro in Brazil.

[3]X-Plane 11 is a modern multiplatform desktop flight simulator developed by Laminar Research.

[4]FlyWithLua is a plugin for X-Plane 11, originally developed by Carsten Lynker, which makes it possible to integrate lua based custom code into the simulator.

A MATLAB code

```
%
%  Code written by Rony Lanssiers in 2018-2019.
%  This work can be shared and adapted as long
%  as appropriate credit is given (CC BY 4.0).
%

clear variables;

fig = figure;
alt = linspace (0, 30000);
plot (alt ./ (6076 * tand (2)), alt ./ 1000, 'k', 'DisplayName', '\alpha = 2 deg');
hold on;
plot (alt ./ (6076 * tand (3)), alt ./ 1000, 'r', 'DisplayName', '\alpha = 3 deg');
plot (alt ./ (6076 * tand (4)), alt ./ 1000, 'b', 'DisplayName', '\alpha = 4 deg');
plot (alt ./ (6076 * tand (5)), alt ./ 1000, 'm', 'DisplayName', '\alpha = 5 deg');

grid on;
legend ('show', 'Location', 'northwest');
xlabel ('distance [Nm]');
xlim ([0 100]);
xticks (0:5:100);
ylabel ('\Delta altitude [\times 1000 ft]');
ylim ([0 30]);
yticks (0:1:30);

print (fig, 'altitudedistance', '-dpng', '-r300');

fig = figure;
gspd = linspace (0, 500);
plot (gspd, gspd .* (6076 * tand (2) / 60), 'k', 'DisplayName', '\alpha = 2 deg');
hold on;
plot (gspd, gspd .* (6076 * tand (3) / 60), 'r', 'DisplayName', '\alpha = 3 deg');
plot (gspd, gspd .* (6076 * tand (4) / 60), 'b', 'DisplayName', '\alpha = 4 deg');
plot (gspd, gspd .* (6076 * tand (5) / 60), 'm', 'DisplayName', '\alpha = 5 deg');

grid on;
legend ('show', 'Location', 'northwest');
xlabel ('ground speed [kts]');
xlim ([0 500]);
xticks (0:25:500);
ylabel ('vertical speed [ft/min]');
ylim ([0 2500]);
yticks (0:100:2500);

print (fig, 'vspeedgspeed', '-dpng', '-r300');
```

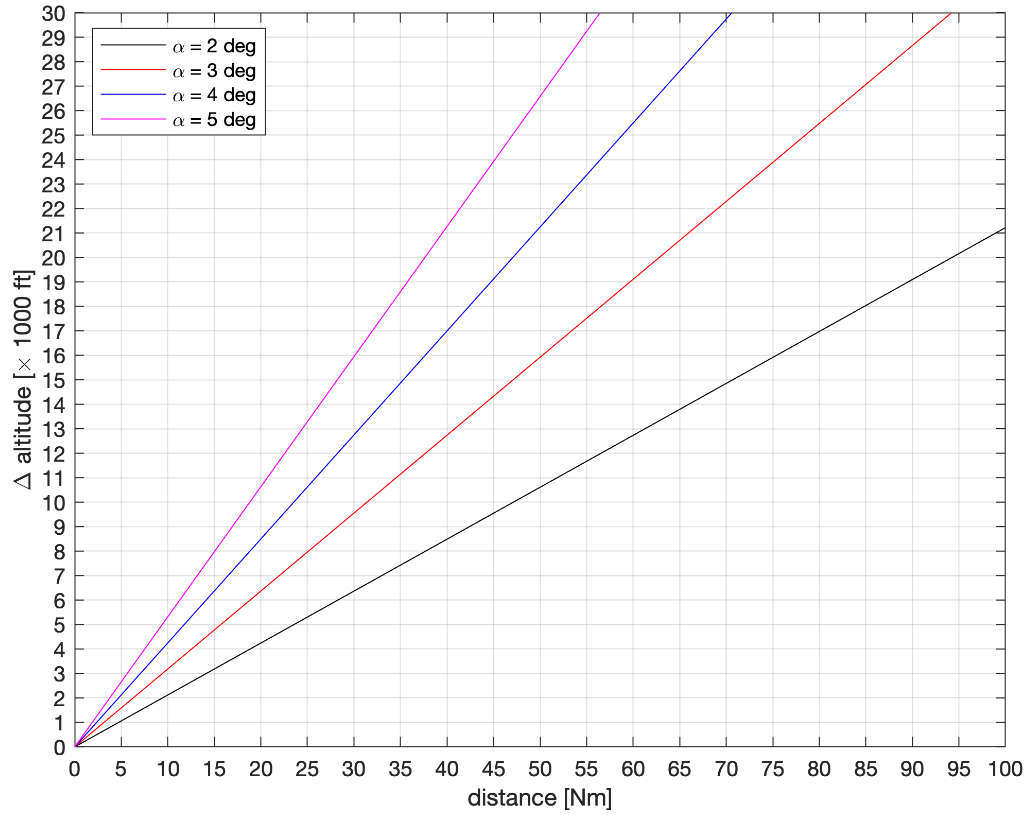
B Lua code

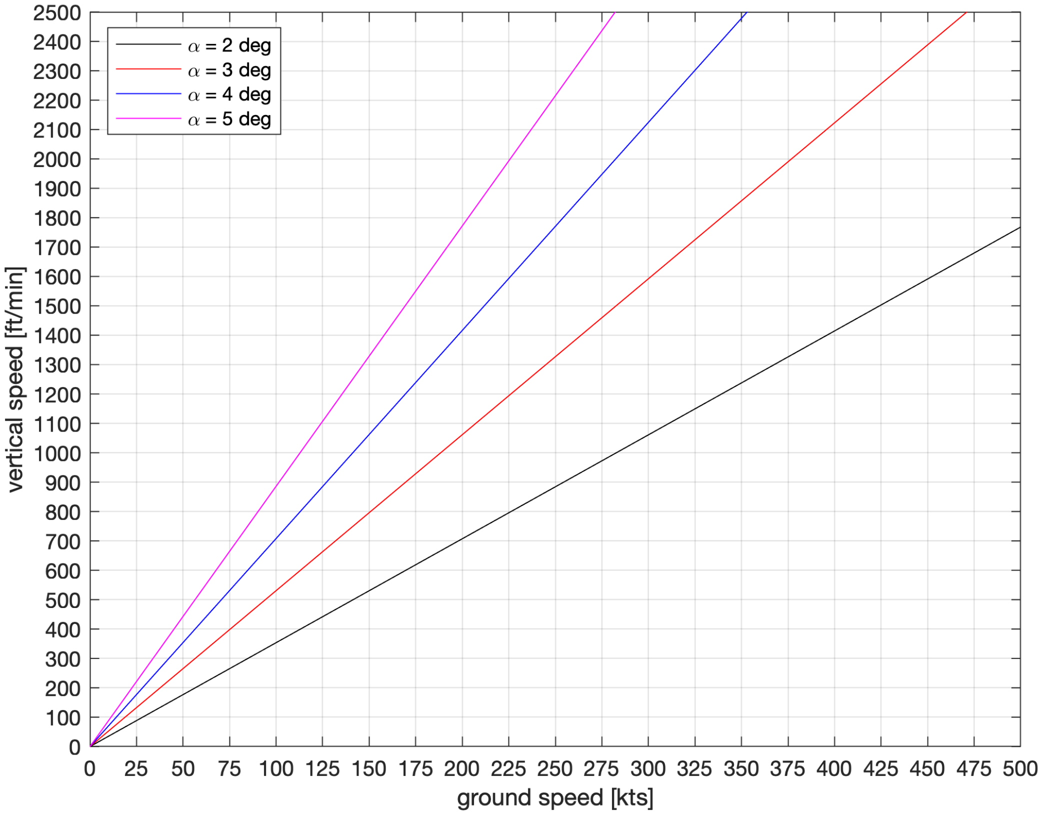
```
--
--  Code written by Rony Lanssiers in 2019.
--  This work can be shared and adapted as long
--  as appropriate credit is given (CC BY 4.0).
--

require("graphics")

dataref ("GSPD", "sim/flightmodel/position/groundspeed", "readonly") -- meter/sec
dataref ("VSPD", "sim/flightmodel/position/vh_ind_fpm", "readonly")  -- feet/min

-- conversion factor kts per m/s
local KTSMS = 1.943844

```

```
-- position and size
local VNAV_X = 30
local VNAV_Y = 700
local VNAV_WIDTH = 20
local VNAV_HEIGHT_POS = 0
local VNAV_HEIGHT_NEG = 600

-- scale in ft/min per pixel
local VNAV_SCALE = 5

-- color settings
local VNAV_RED = 0.0
local VNAV_GREEN = 1.0
local VNAV_BLUE = 0.0
local VNAV_ALPHA = 0.6

-- show/hide toggle
VNAV_SHOWGAUGE = false

function VNAV_DrawMarker (gsangle)
  vspd = 101.3 * math.tan (gsangle / 180.0 *  math.pi) * GSPD * KTSMS
  y = VNAV_Y + vspd / VNAV_SCALE
  if y <= VNAV_Y + VNAV_HEIGHT_POS and y >= VNAV_Y - VNAV_HEIGHT_NEG then
    graphics.set_color (VNAV_RED, VNAV_GREEN, VNAV_BLUE, 1.0)
    graphics.draw_line (VNAV_X - 5, y, VNAV_X + VNAV_WIDTH + 5, y)
    draw_string (VNAV_X - 20, y - 3, string.format ("%i", gsangle), VNAV_RED, VNAV_GREEN, VNAV_BLUE)
    draw_string (VNAV_X + VNAV_WIDTH + 10, y - 3, string.format ("%i", vspd), VNAV_RED, VNAV_GREEN,
     VNAV_BLUE)
  end
end

function VNAV_DrawGauge ()
  if VNAV_SHOWGAUGE then
    y = VNAV_Y + VSPD / VNAV_SCALE
    graphics.set_color (VNAV_RED, VNAV_GREEN, VNAV_BLUE, VNAV_ALPHA)
    graphics.draw_rectangle (VNAV_X, VNAV_Y, VNAV_X + VNAV_WIDTH, math.max (math.min (y, VNAV_Y +
     VNAV_HEIGHT_POS), VNAV_Y - VNAV_HEIGHT_NEG))

    graphics.set_color (VNAV_RED, VNAV_GREEN, VNAV_BLUE, 1.0)
    graphics.draw_line (VNAV_X - 5, VNAV_Y, VNAV_X + VNAV_WIDTH + 5, VNAV_Y)
    graphics.draw_line (VNAV_X + VNAV_WIDTH / 2, VNAV_Y + VNAV_HEIGHT_POS, VNAV_X + VNAV_WIDTH / 2,
     VNAV_Y - VNAV_HEIGHT_NEG)

    VNAV_DrawMarker (-2)
    VNAV_DrawMarker (-3)
    VNAV_DrawMarker (-4)
    VNAV_DrawMarker (-5)
  end
end

do_every_draw ("VNAV_DrawGauge ()")

function VNAV_ToggleGauge ()
  if VNAV_SHOWGAUGE then
    VNAV_SHOWGAUGE = false
  else
    VNAV_SHOWGAUGE = true
  end
end

create_command ("RLA/VNAV/ToggleGauge", "Toggle VNAV Gauge", "VNAV_ToggleGauge ()", "", "")
```

C Quick reference page

This appendix reprints figures 4 and 6 without captions on a single page for easy reference. The reader remains responsible for verifying their correctness and correct application.


α = 2 deg
α = 3 deg
α = 4 deg
α = 5 deg
Δ altitude [× 1000 ft]
distance [Nm]


α = 2 deg
α = 3 deg
α = 4 deg
α = 5 deg
vertical speed [ft/min]
ground speed [kts]