```matlab
% This program calculates how much you should turn your aircraft to
% compensate for the wind. The code is written in MATLAB by FlygFan.

% Some MATLAB expressions: if we want to find sin, cos or tan of an
% angle using degrees, we type sind, cosd, tand. If we want to find
% arcsin, arccos or arctan we type asind, acosd, atand (in degrees).

pr = app.pr.Value; % pr is the desired heading of the aircraft.
% "app.pr.Value" is the heading value of the input from the user.
vr = app.vr.Value; % vr is the wind direction. "app.vr.Value" is the
% direction value of the input from the user.
ph = app.ph.Value/3.6; % ph is the indicated air speed (IAS) of the
% aircraft. "app.ph.Value" is the speed value of the input from the user
% in km/h. We divide with 3.6 to convert to m/s from km/h.
ph = ph + ph*app.alt.Value*4.92126e-5; % Converts IAS to TAS (TAS
% increases with 0.00492126% per meter (equivalent to 1.5% per
% 1000ft)). "app.alt.Value" is the altitude value of the input from
% the user.
vh = app.vh.Value; % vh is the wind speed in m/s. "app.vh.Value" is
% the speed value of the input from the user.
cr = app.cr.Value; % cr is the climb rate of the aircraft.
% "app.cr.Value" is the climb rate value of the input from the user.

% I am assuming the wind only travels horizontally. Therefore we only
% want the horizontal speed of the aircraft.
if cr > 0 || cr < 0  % If the aircraft is climbing.
    ph = cosd(asind(cr/ph))*ph; % This line calculates the horizontal
% speed of the aircraft given its climbrate.
end

gs = (ph - vh*cosd(pr - vr))*0.06 % gs is the ground speed of the aircraft in km/min.
app.gs.Value = round(gs,1) % Displays the ground speed in the "Ground Speed" text box

% In the following if-statements, the program finds the relative angle
% beta between the aircraft and the wind.
% This angle has a minimum value of 0 and a maximum value of 90.
% For instance, the relative angle between one wind coming from 80
% degrees and another from 100 degrees between an aircraft traveling in
% 0 degrees is the same - 80 degrees.
% The program also figures out wheter we should compensate to the left
% or the right, or nothing at all.
if pr == vr || (abs(vr - pr) == 180 || abs(vr-pr) == 360) % If the
% relative angle is 0 or 180 degrees we should not turn left/right.
    beta = 0;
    svar2 = '';
elseif pr > vr
    beta = pr-vr;
    svar2 = 'left';
    if beta > 90
        beta = vr + 180 - pr;
        if beta > 0
            svar2 = 'left';
```

```
        else
            svar2 = 'right';
        end
        beta = abs(vr + 180 - pr);
        if beta > 90
            beta = beta - 2*abs(90-beta);
            svar2 = 'right';
        end
    end
elseif vr > pr
    svar2 = 'right';
    beta = vr - pr;
    if beta > 90
        beta = pr + 180 - vr;
        if beta > 0
            svar2 = 'right';
        else
            svar2 = 'left';
        end
        beta = abs(pr + 180 - vr);
        if beta > 90
            beta = beta - 2*abs(90-beta);
            svar2 = 'left';
        end
    end
end

% We have found the relative angle beta between the aircraft's desired
% heading and the wind direction. Now we need to find how much to
% compensate for the wind. This is done numerically as follows:
% alpha_next are the various angles between the desired heading and
% beta, where one of these angles is the correct one to compensate
% with. The following for-loop goes through these various angles and
% calculates the resulting angle alpha3 between the heading of the
% aircraft (when affected by the wind) and the desired heading. By
% putting the desired heading to 0 degrees, we want alpha_next to be
% such an angle that when the aircraft is flying with the angle
% alpha_next, the angle between the resulting heading alpha3 and the
% desired heading is as close to 0 as possible. When this value is
% found we can get the value of alpha_next which is the correct angle
% to compensate with.
alpha3 = [];
for alpha_next = 0:0.01:beta/2.2 % From the limited values of
    % airspeed, wind speed etc. the maximum angle to compensate with
    % is roughly beta/2.2, so by limiting alpha_next (by dividing
    % with 2.2) we remove unnecessary required computing power.
    ph_orto = ph * cosd(alpha_next); % We only want the
    % orthogonal/perpendicular air speed compared to the wind.
    alpha2 = atand(sind(beta-alpha_next)*vh/ph_orto); % Middle step to
    % find alpha3.
    alpha3 = [alpha3,alpha_next - alpha2]; % Vector which contains the
    % various values of alpha3 when we change alpha_next.
end
```

```
% In the following 3 rows we find the correct value for alpha_next.
alpha_next = 0:0.01:beta/2.2;
[vinkel,indx] = min(abs(alpha3)); % We find which angle, in the
% alpha3 vector, that has the value closest to 0 since that angle was
% the correct angle to compensate with.
alphares = alpha_next(indx); % alphares is the angle we are
% compensating with.

%this is statement keeps the sign of the compensation
if strcmp(svar2, 'right')
    app.dispVk.Value = -round(alphares,1); % this displays alphares in the "Compensation" text
box
else
    app.dispVk.Value = round(alphares,1);
end

% The following if-statements finds the new heading given which way we
% should turn and the value of alphares.
if strcmp(svar2,'right')
    nyrikt = pr + alphares;
    if nyrikt > 360
        nyrikt = nyrikt - 360;
    end
elseif strcmp(svar2,'left')
    nyrikt = pr - alphares;
    if nyrikt < 0
        nyrikt = 360 + nyrikt;
    end
elseif strcmp(svar2,'')
        nyrikt = pr;
end

app.nyrikt.Value = round(nyrikt,1); % Displays the new heading in the
% "New Heading" text box.
```

Published with MATLAB® R2018a