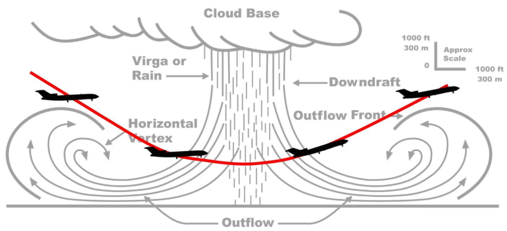
Aircraft Go-Around in the Presence of Windshear

rem: `Aircraft Go-Around in the Presence of Windshear`

This problem is taken from the examples implemented in ICLOCS2 (www.ee.ic.ac.uk/ICLOCS/). The description is reproduced here below.

The problem was initially presented by [1-2]. This implementation contain modifications to the original formulation by [3].

Consider following problem where a commercial aircraft encountered windshear during landing and need to go-around.



The objective is to maximize the lowest altitude ever reached

$$\max h_{min}$$

subject to dynamics constraints

$$\begin{aligned}
\dot{d} &= v\cos(\gamma) + w_d(d)\\
\dot{h} &= v\sin(\gamma) + w_h(d,h)\\
\dot{v} &= \frac{1}{m}\Big(T(v)\cos(\alpha+\delta) - D(v,\alpha)\Big) - g\sin(\gamma) - \dot{w}_d(d,\dot{d}\,)\cos(\gamma) - \dot{w}_h(d,h,\dot{d}\,,\dot{h})\sin(\gamma)\\
\dot{\gamma} &= \frac{1}{mv}\Big(T(v)\sin(\alpha+\delta) + L(v,\alpha)\Big) - \frac{g}{v}\cos(\gamma) + \frac{1}{v}\dot{w}_d(d,\dot{d}\,)\sin(\gamma) - \frac{1}{v}\dot{w}_h(d,h,\dot{d}\,,\dot{h})\cos(\gamma)
\end{aligned}$$

path constraint

$$h \geq h_{min}$$

simple bounds on variables

$$\begin{gathered}
0 \leq d \leq 10000 \text{ [ft]}\\
0 \leq h \leq 1000 \text{ [ft]}\\
0 \leq v \leq +\infty \text{ [ft/s]}\\
-\infty \leq \gamma \leq +\infty \text{ [deg]}\\
-17 \leq \alpha \leq 17 \text{ [deg]}\\
-3 \leq \dot{\alpha} \leq 3 \text{ [deg/s]}
\end{gathered}$$

and boundary conditions

$$\begin{gathered}
d(0) = 0 \text{ [ft]}, h(0) = 600 \text{ [ft]}, v(0) = 239.7 \text{ [ft/s]}, \gamma(0) = -2.25 \text{ [deg]}, \alpha(0) = 7.35 \text{ [deg]}\\
\gamma(t_F) = 7.43 \text{ [deg]}
\end{gathered}$$

where $d(t), h(t), v(t), \gamma(t), \alpha(t)$ stand for position [ft], altitude [ft], speed [ft/s], flight path angle [rad] and angle of attack [rad] respectively. The latter is the control variable.

References:

1. R. Bulirsch, F. Montrone, and H. Pesch, *Abort landing in the presence of windshear as a minimax optimal control problem, part 1: Necessary conditions*, Journal of Optimization Theory and Applications, 70(1), pp 1-23, 1991.
2. R. Bulirsch, F. Montrone, and H. Pesch, *Abort landing in the presence of windshear as a minimax optimal control problem, part 2: Multiple shooting and homotopy*, Journal of Optimization Theory and Applications, 70(2), pp 223-254, 1991.
3. J. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming: Second Edition*, Advances in Design and Control, Society for Industrial and Applied Mathematics, 2010.
4. P. Falugi, E. Kerrigan, E. van Wyk, *Imperial College London Optimal Control Software User Guide (ICLOCS)*, http://www.ee.ic.ac.uk/ICLOCS/user_guide.pdf

```
run:
  snopt.opt < major optimality tolerance 5e-3
  model-option = optfile = 1
  default-time-steps-number = 12

par:
  beta_0 = 0.3825
  beta_0_dot = 0.2
  A_0 = 0.4456e+5
  A_1 = -0.2398e+2
  A_2 = 0.1442e-1
  B_0 = 0.15523333333
  B_1 = 0.1236914764
  B_2 = 2.420265075
  C_0 = 0.7125
  C_1 = 6.087676573
  C_2 = -9.027717451
  mg = 150000
  g = 32.172
  delta = deg2rad(2)
  rho = 0.2203e-2
  S = 1560
  alpha_star = deg2rad(12)
  alpha_max = deg2rad(17)
  m = mg / g
  tf = 40

fun:
  $wd = PolyDX
     0 -4e-11        6e-8         0            0           -50
   500  0            0            0            0.025       -45
  4100  4e-11       -2e-8        -3e-5         0.025        45
  4600  0            0            0            0            50

  $wh = PolyDX
     0  6.2808e-11  -8.0288e-08   0.0000e+00   0.0000e+00   0.0000e+00
   500  0.0000e+00   2.1622e-08  -3.4949e-05  -3.1127e-02  -6.1105e+00
   700  0.0000e+00   1.6377e-07  -6.7702e-05  -4.2512e-02  -1.3561e+01
   900  0.0000e+00   1.4608e-07  -3.1452e-05  -4.9941e-02  -2.3461e+01
  1100  0.0000e+00   1.0690e-07   3.8109e-07  -4.4992e-02  -3.3539e+01
  1300  0.0000e+00   6.9672e-08   1.4096e-05  -3.2012e-02  -4.1667e+01
  1500  0.0000e+00   4.2736e-08   1.3175e-05  -1.8013e-02  -4.6948e+01
  1700  0.0000e+00   2.3500e-08   6.9864e-06  -7.6143e-03  -4.9681e+01
  1900  0.0000e+00   7.8226e-09   2.2661e-06  -1.9997e-03  -5.0737e+01
  2100  0.0000e+00   2.5829e-10   3.0895e-07  -1.5457e-04  -5.0984e+01
  2300  0.0000e+00  -2.5829e-10   4.6393e-07   0.0000e+00  -5.1000e+01
  2500  0.0000e+00  -7.8226e-09   6.9597e-06   1.5457e-04  -5.0984e+01
  2700  0.0000e+00  -2.3500e-08   2.1086e-05   1.9997e-03  -5.0737e+01
  2900  0.0000e+00  -4.2736e-08   3.8816e-05   7.6143e-03  -4.9681e+01
  3100  0.0000e+00  -6.9672e-08   5.5899e-05   1.8013e-02  -4.6948e+01
  3300  0.0000e+00  -1.0690e-07   6.4520e-05   3.2012e-02  -4.1667e+01
  3500  0.0000e+00  -1.4608e-07   5.6197e-05   4.4992e-02  -3.3539e+01
  3700  0.0000e+00  -1.6377e-07   3.0557e-05   4.9941e-02  -2.3461e+01
  3900  0.0000e+00  -2.1622e-08  -2.1976e-05   4.2512e-02  -1.3561e+01
  4100  6.2808e-11  -4.5329e-08  -2.6220e-05   2.8812e-02  -6.1105e+00
  4600  0.0000e+00   0.0000e+00   0.0000e+00   0.0000e+00   1.0000e-15

var:
  hmin

dyn:
  pos
  h
  v
```

```
  fpa
  alpha

lim:
  0 <= pos <= 10000
  100 <= h <= 1000
  0.01 <= v <= 300
  deg2rad(-180) <= fpa <= deg2rad(180)
  -alpha_max <= alpha <= alpha_max

t=t0:
  pos = 0              # Initial position [ft]
  h = 600              # Initial altitude [ft]
  v = 239.7            # Initial speed [ft/s]
  fpa = deg2rad(-2.25)  # Initial flight path angle [rad]
  alpha = deg2rad(7.35) # Initial angle of attack [rad]

t=tf:
  fpa = deg2rad(7.43)   # Final flight path angle [rad]

ini:
  pos = linspace(0,900)
  h = initial(h)
  v = initial(v)
  fpa = linspace(initial(fpa),final(fpa))
  alpha = initial(alpha)
  hmin = 502

exp:
  beta(x) == ifthen(x < (1-beta_0)/beta_0_dot, beta_0_dot*x+beta_0, 1)
  cl_1(x) == c_0 + c_1*x
  cl_2(x) == c_0 + c_1*x + c_2*sqr(x-alpha_star)
  cl_p(x) == ifthen(x < alpha_star, cl_1(x), cl_2(x))
  T == beta(Time) * (A_0 + A_1*v + A_2*v*v)
  D == 0.5 * (B_0 + B_1*alpha + B_2*alpha*alpha) * rho * S * v * v
  L == 0.5 * cl_p(alpha) * rho * S * v * v
  wd == $wd(pos)
  wh == $wh(pos) * h/1000
  # wd_dot == $wd(pos+pos_dot) - $wd(pos)
  # wh_dot == $wh(pos+pos_dot) * (h+h_dot)/1000 - $wh(pos) * h/1000
  wd_dot == slope(wd)
  wh_dot == slope(wh)
  pos_dot == v*cos(fpa) + wd
  h_dot == v*sin(fpa) + wh

equ:
  pos´ == pos_dot
  h´ == h_dot
  v´ == T/m*cos(alpha+delta) - D/m - g*sin(fpa) - wd_dot*cos(fpa) - wh_dot*sin(fpa)
  fpa´ == T/m/v*sin(alpha+delta) + L/m/v - g/v*cos(fpa) + wd_dot*sin(fpa)/v - wh_dot*cos(fpa)/v
  h >= hmin
  deg2rad(-3) <= slope(alpha) <= deg2rad(3)
  alpha :: Spline3

obj:
  maximize hmin using dnlp with snopt
```