



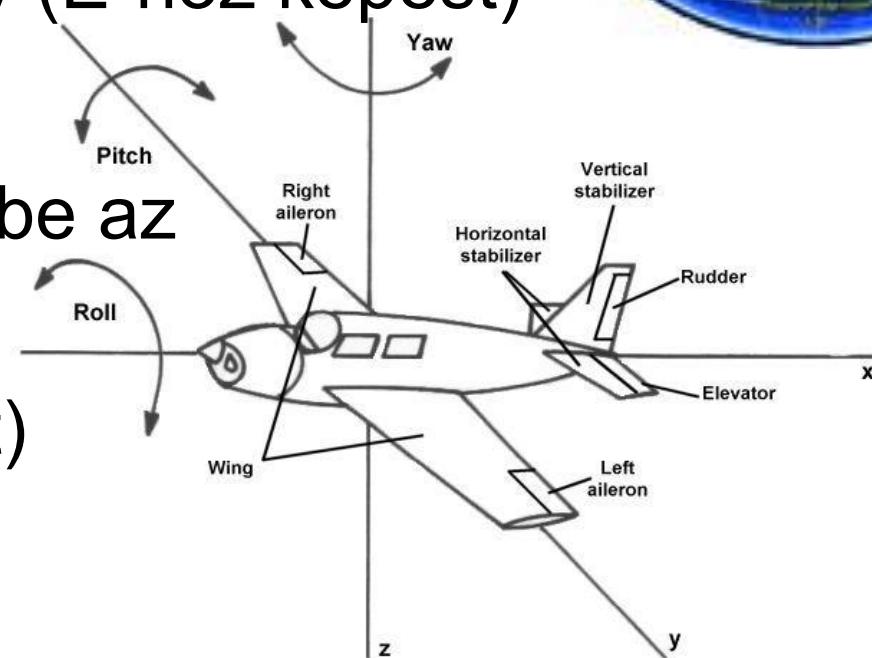
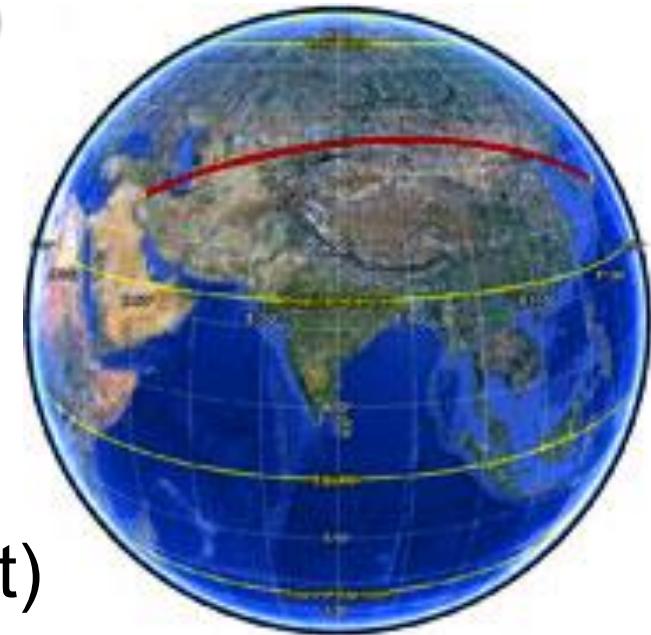
Robotrepülőgépek útvonaltervezése

Intelligens Rendszerek
Gyakorlat



Navigáció

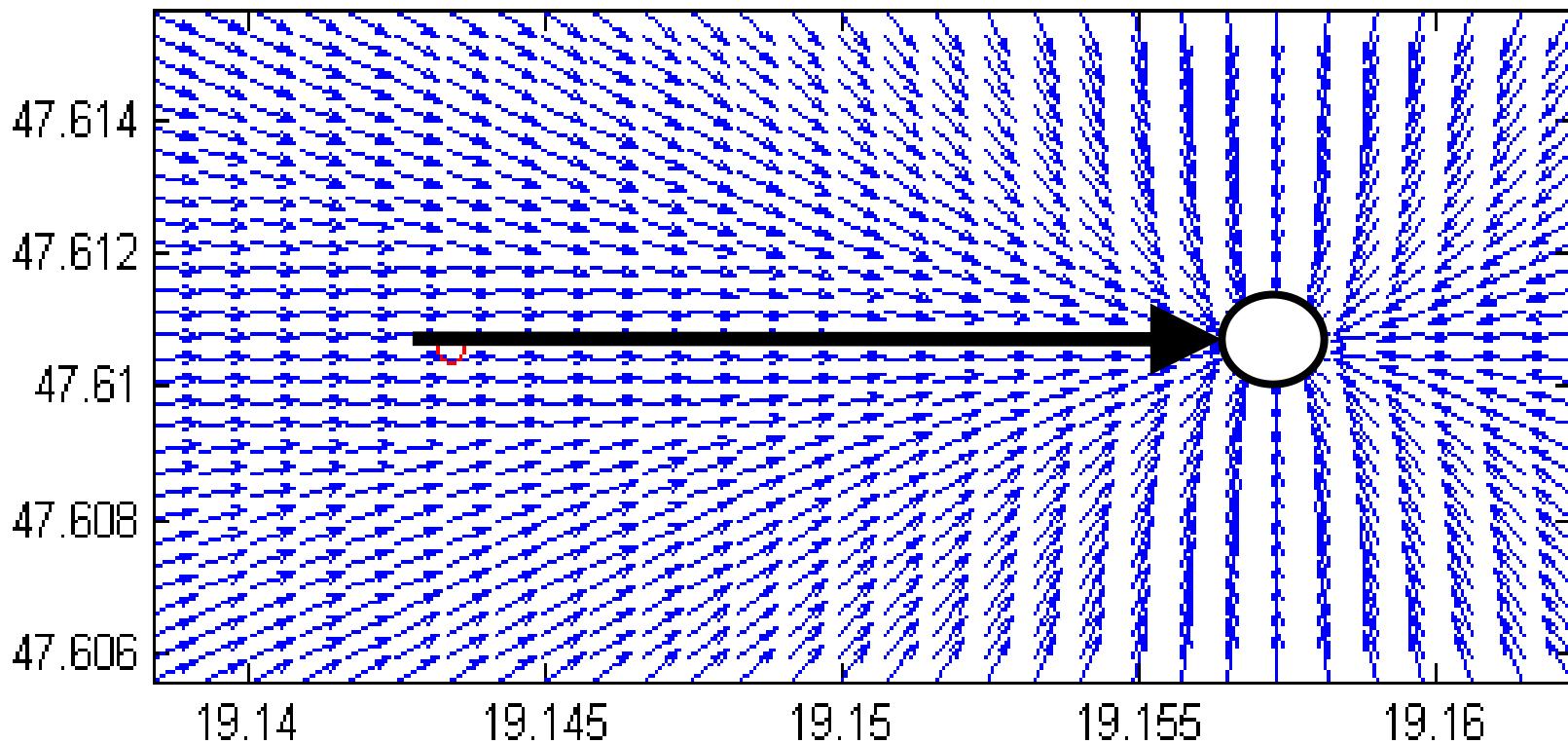
- Course
 - Tervezett útvonal
- Bearing
 - Haladási irány (É-hoz képest)
- Heading
 - A gép orra ebbe az irányba néz (É-hoz képest)





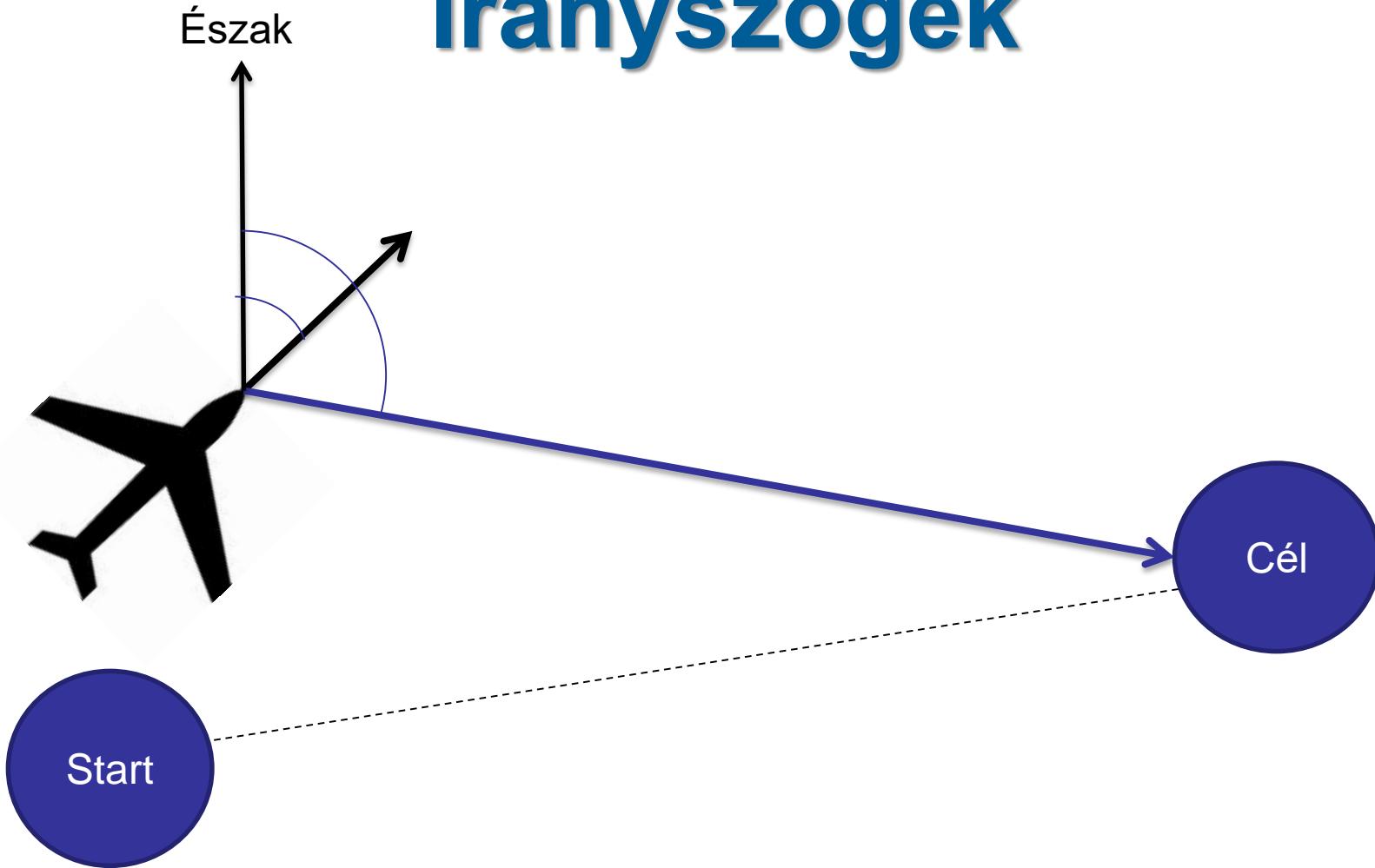
Fordulópontok

- A gép mindenkor egy koordináta párba álló pozíció felé halad





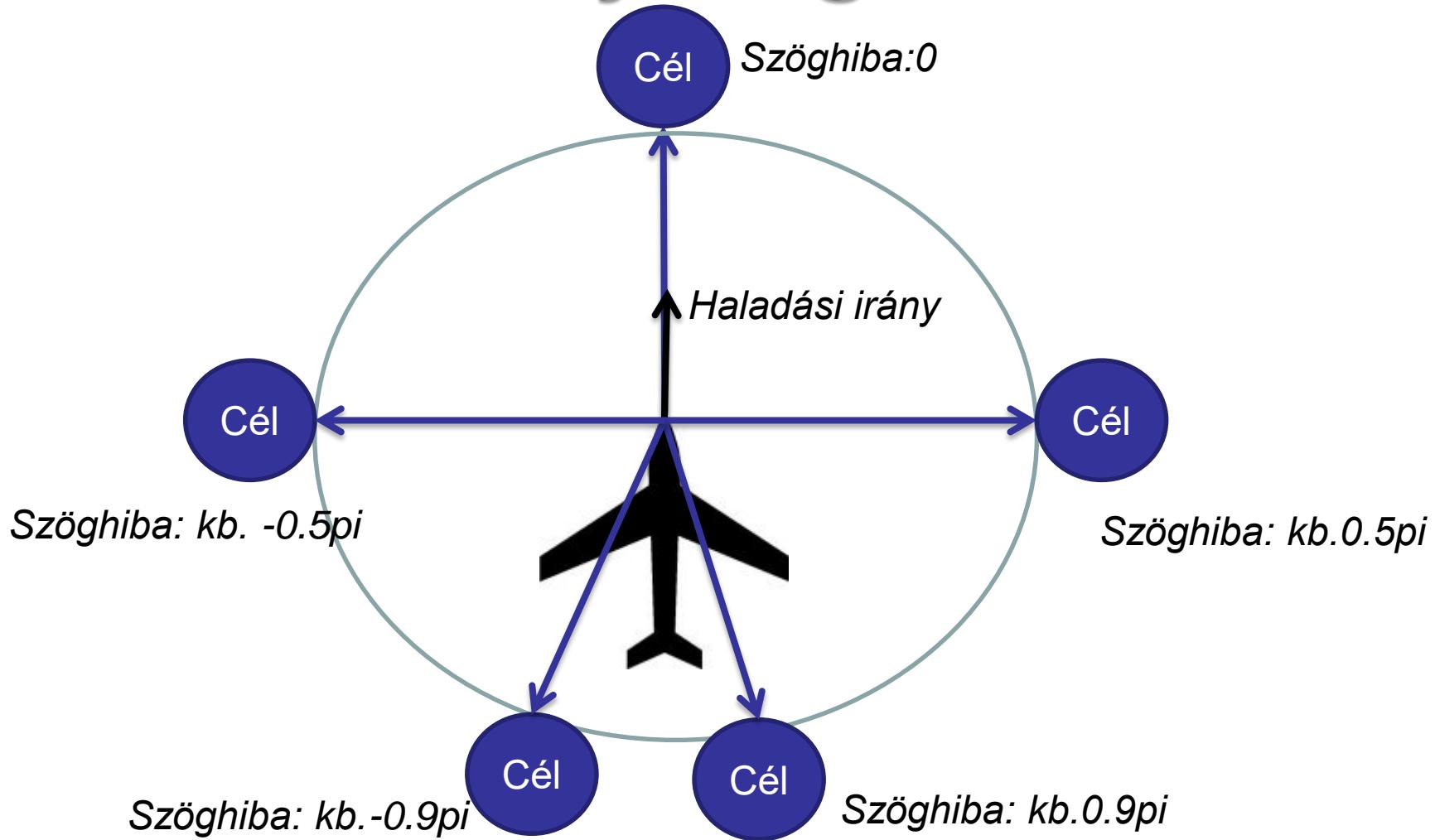
Irányszögek



(Szöghiba: -pi és +pi között!)



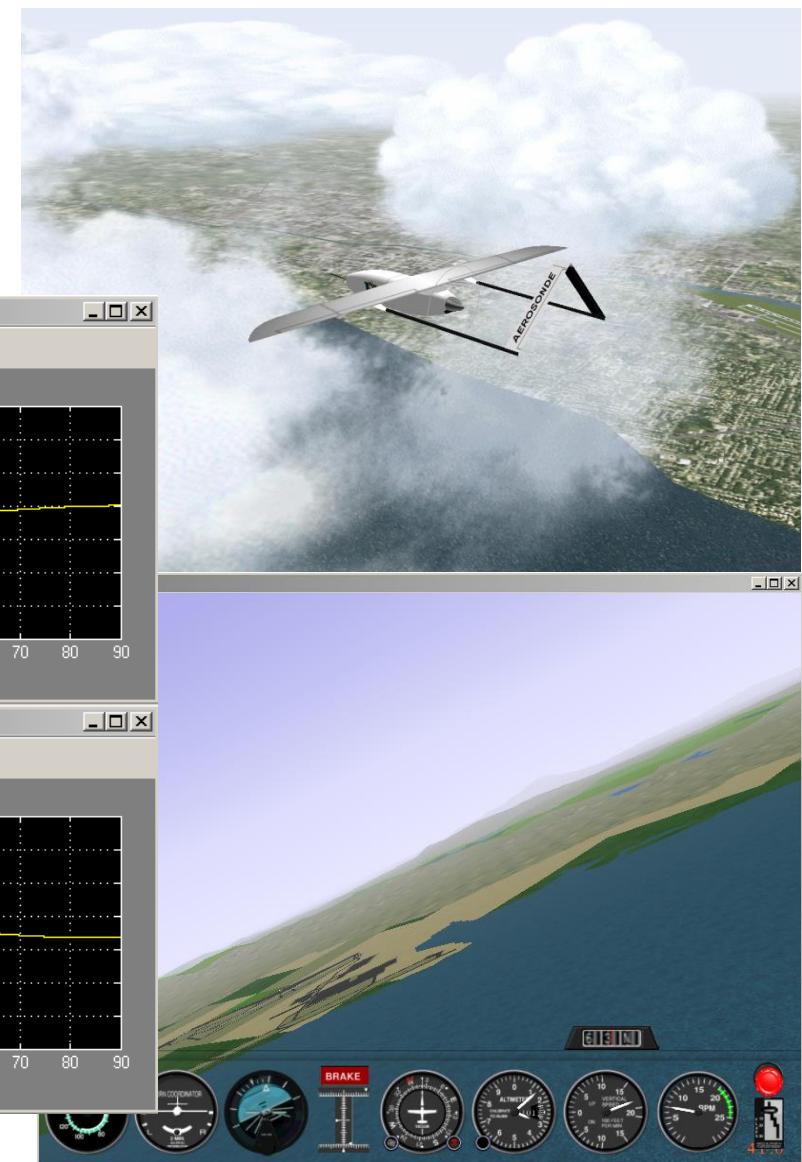
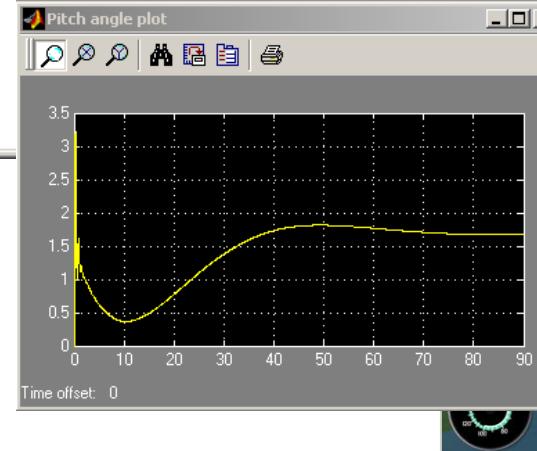
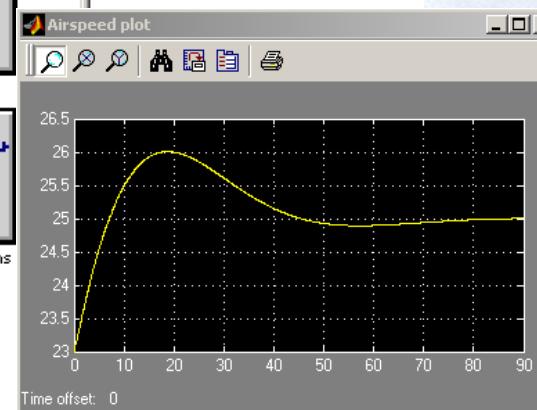
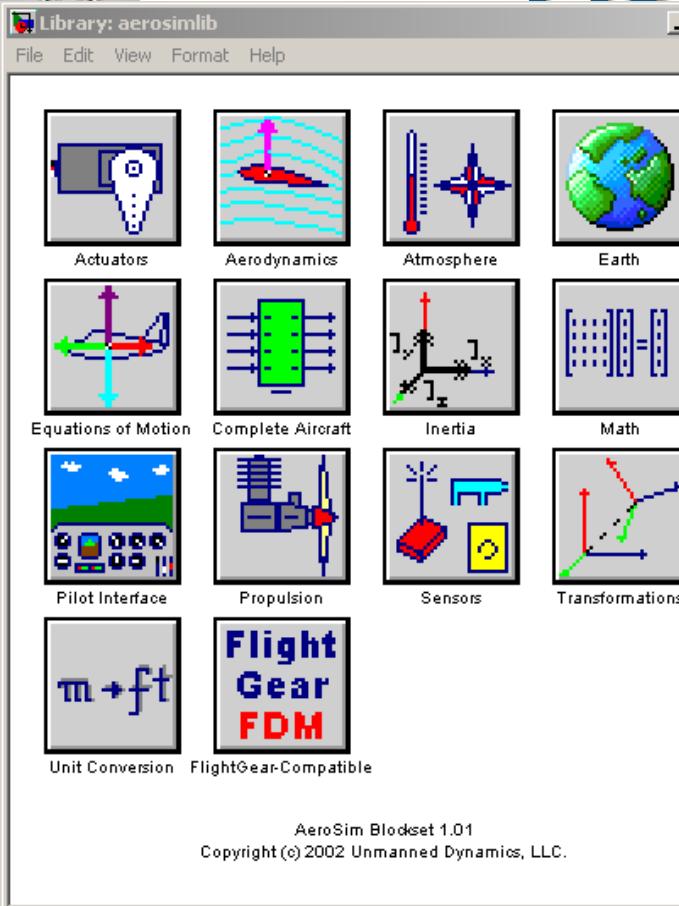
Irányszögek



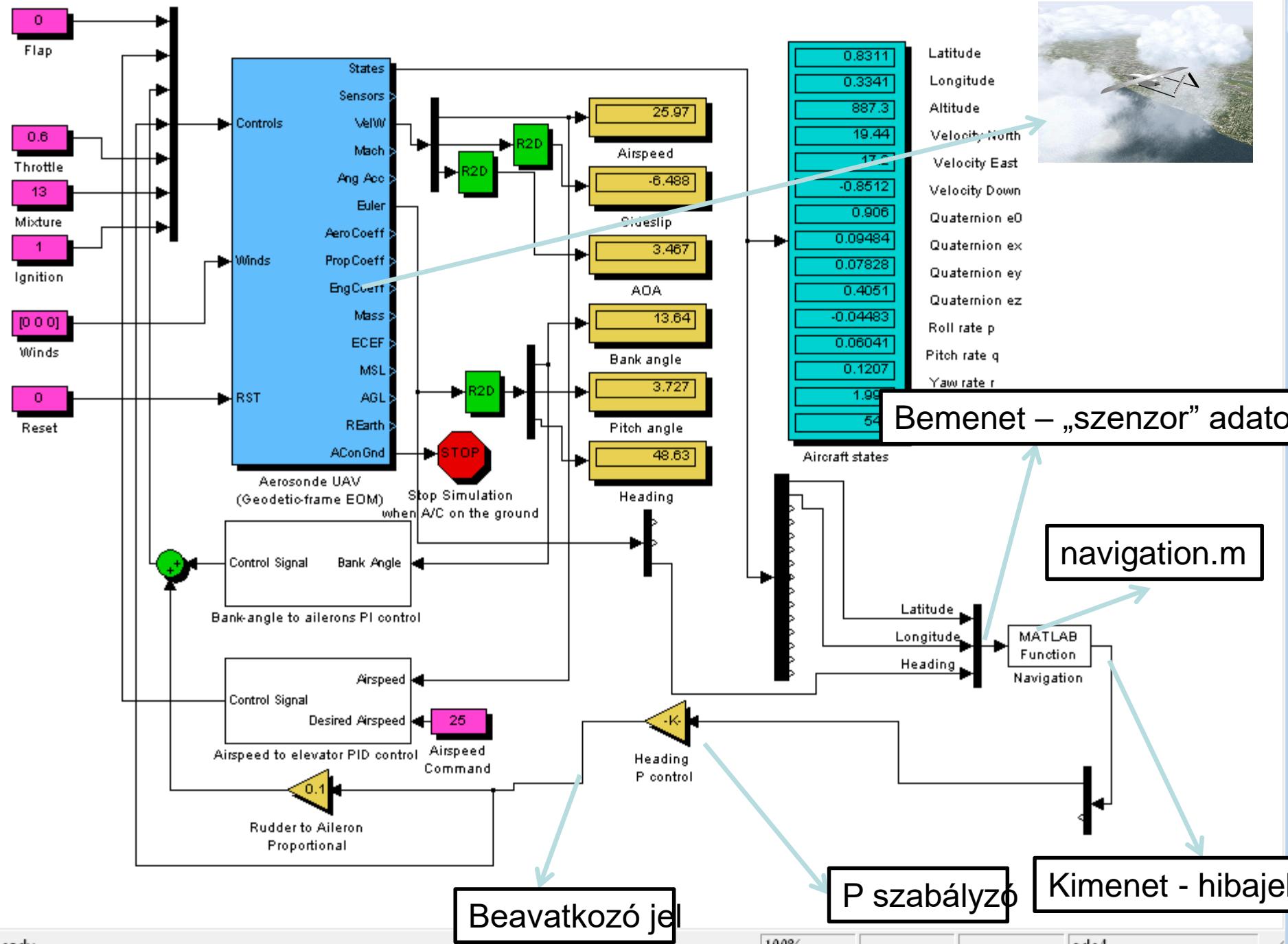
(Szöghiba: $-pi$ és $+pi$ között!)



Aersim Blockset



E
T
E
M



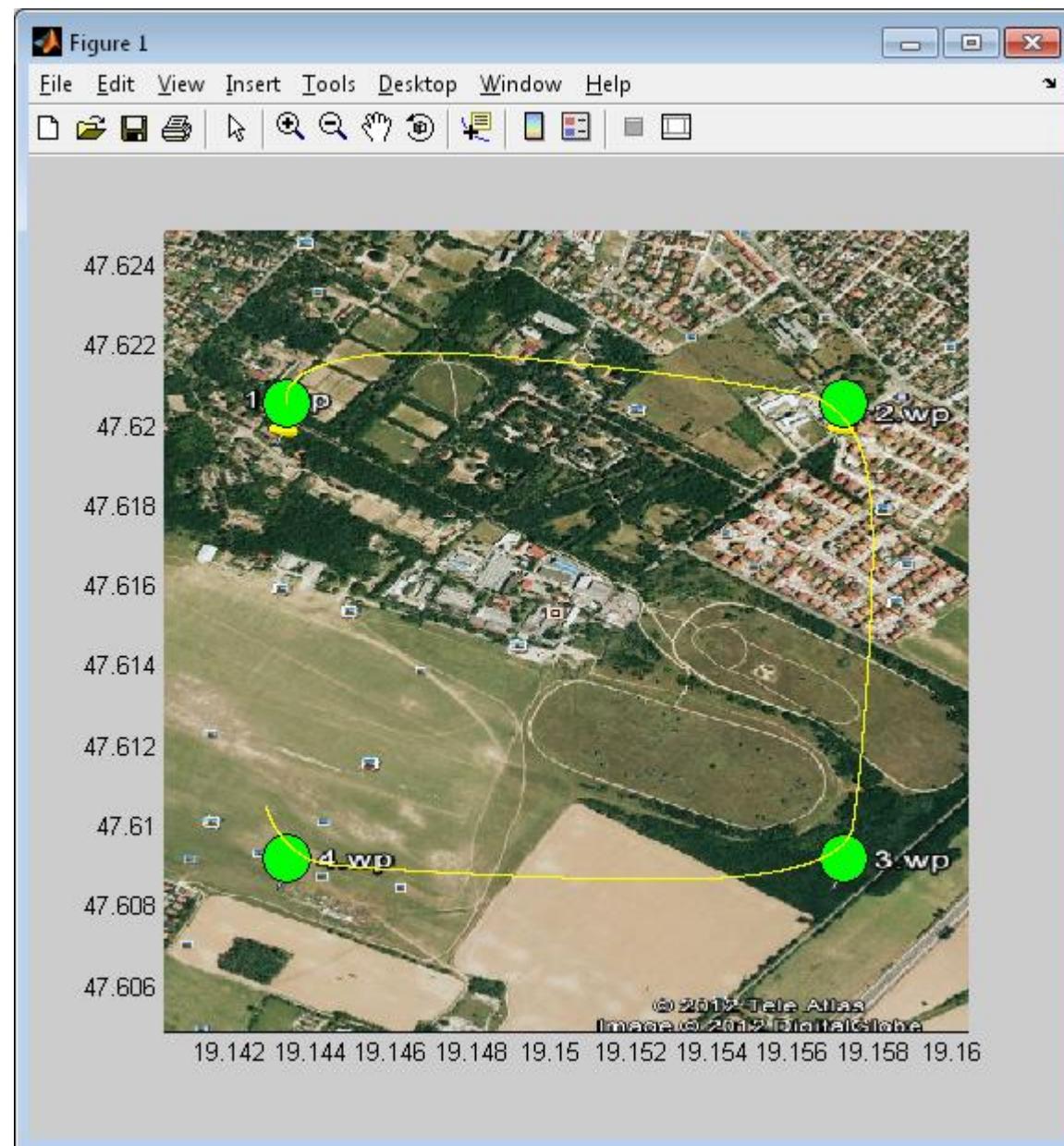


Inicializálás (init.m)

```
1 %INIT
2 %!!!!!! RUN THIS FIRST !!!!!
3 clear all;
4 close all;
5
6 global target_waypoint
7 target_waypoint=2;    %Cél fordulópont sorszáma
8
9 global lat;
10 global lon;
11 lat=1;
12 lon=2;
13
14 %Dunakeszi reptér 4 * (kb)1000m
15 global waypoint_list
16 waypoint_list=[  
17     47.6206, 19.1434; %1. fordulópont 'Decimal Degree' formátumban  
18     47.6206, 19.1573; %2. fordulópont 'Decimal Degree' formátumban  
19     47.6092, 19.1573; %3. fordulópont 'Decimal Degree' formátumban  
20     47.6092, 19.1434 ];%4. fordulópont 'Decimal Degree' formátumban
21 waypoint_list=deg2rad(waypoint_list)
22 %1-től számoz, nem nullától!!!
23 %waypoint_list(1,lat) : 47.6206
24 %waypoint_list(1,lon) : 19.1434
25
26 plotMap();
27
```



Futtatás





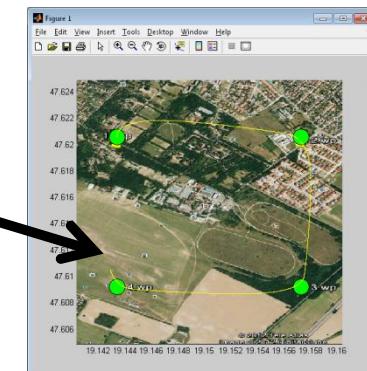
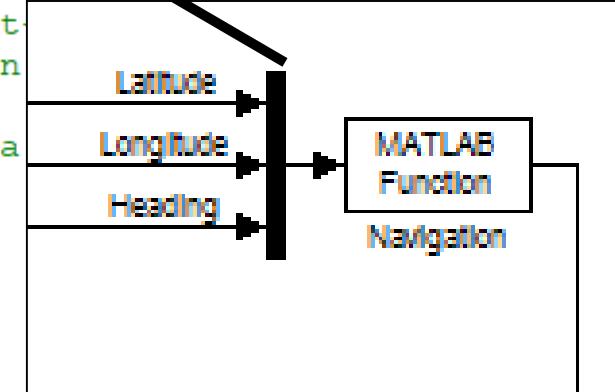
Navigation.m

```
1 function [ output_args ] = navigation( input_args )
2
3 %globális változók behívása
4 global target_waypoint; %cél fordulópont sorszáma a listán belül
5 global lat; % 1-es index (latitude) waypoint_list-hez
6 global lon; % 2-es index (longitude) waypoint_list-hez
7 global waypoint_list; % fordulópont lista (lat,lon koordináta párok)
8 global h; % plot handler
9 waypoint_radius=100; %[méter] fordulópont rádiusza
10
11 %input argomentumok átadása
12 uav_lat=input_args(1);
13 uav_lon=input_args(2);
14 uav_heading=input_args(3);
15
16 hold on; %diagram eddig kirajzolt elemeinek a megtartása
17 plot(rad2deg(uav_lon), rad2deg(uav_lat),'--y'); % aktuális pozíció kirajzolása
18
```



Navigation.m

```
1 function [ output_args ] = navigation( input_args )  
2  
3 %globális változók behívása  
4 global target_waypoint; %cél fordulópont sorszáma a listán belül  
5 global lat; % 1-es index (latitude) waypoint_list-hez  
6 global lon; % 2-es index (longitude) waypoint_list-  
7 global waypoint_list; % fordulópont lista (lat,lon)  
8 global h; % plot handler  
waypoint_radius=100; %[méter] fordulópont rádiusa  
9  
10 %input argumentumok átadása  
11 uav_lat=input_args(1);  
12 uav_lon=input_args(2);  
13 uav_heading=input_args(3);  
14  
15  
16 hold on; %diagram eddig kirajzolt elemeinek a megtartása  
17 plot(rad2deg(uav_lon), rad2deg(uav_lat), '--y'); % aktuális pozíció kirajzolása  
18
```





A feladat:

```
22
23 % 1; adott pontban (uav) a kívánt haladási irány kiszámítása
24 %bearing.m-et felhasználva -> bearing([FROM_lat FROM_lon TO_lat TO_lon]); tömbként
25 megadva.
26 desired_bearing=0;
27
28 % 2; uav - cél fordulópont távolság számítása
29 %distance.m-et felhasználva -> distance([FROM_lat FROM_lon TO_lat TO_lon]); tömbként
30 megadva. Távolság méterben értendő.
31 waypoint_distance=0;
32
33 % 3; fordulópont léptetése, ha a fordulópont távolsága a fordulópont
34 % rádiusnál kisebb
35 target_waypoint=2;
36
37 % 4. szöghiba (kívánt és valós haladási irány különbsége, -Pi...+Pi (-180...+180°)
38 között megadva)
39
40 heading_error=0;
41
42 %kimenet
43 output_args=[heading_error, waypoint_distance/1000]; % szöghiba radiánban és távolság
44 km-ben
```