

Intelligens Rendszerek

Robotok 4.

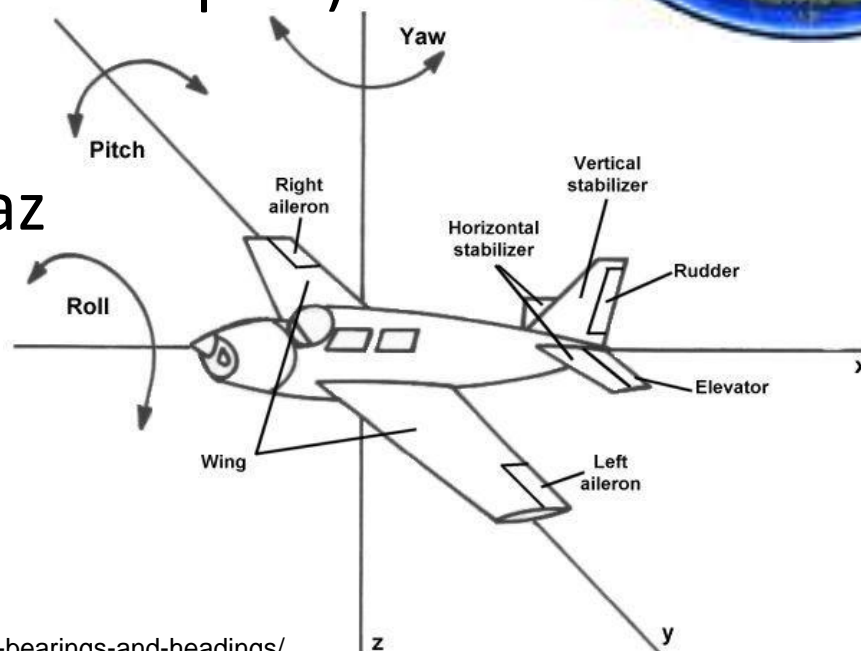
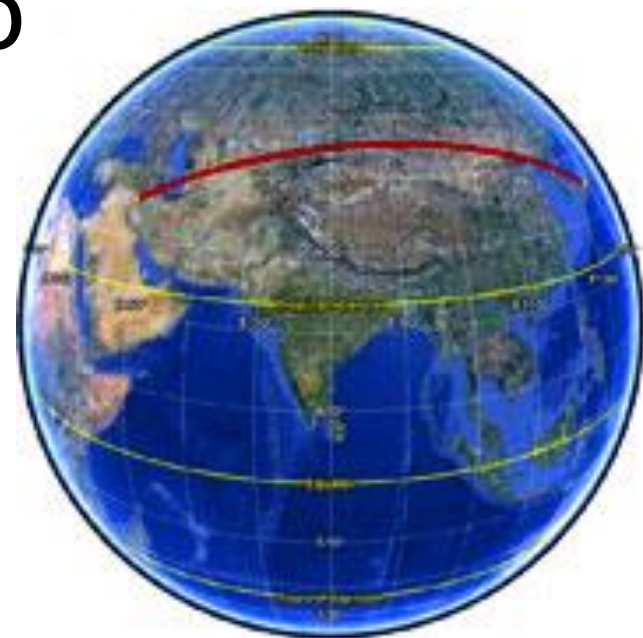
<http://mobil.nik.uni-obuda.hu>

UAV



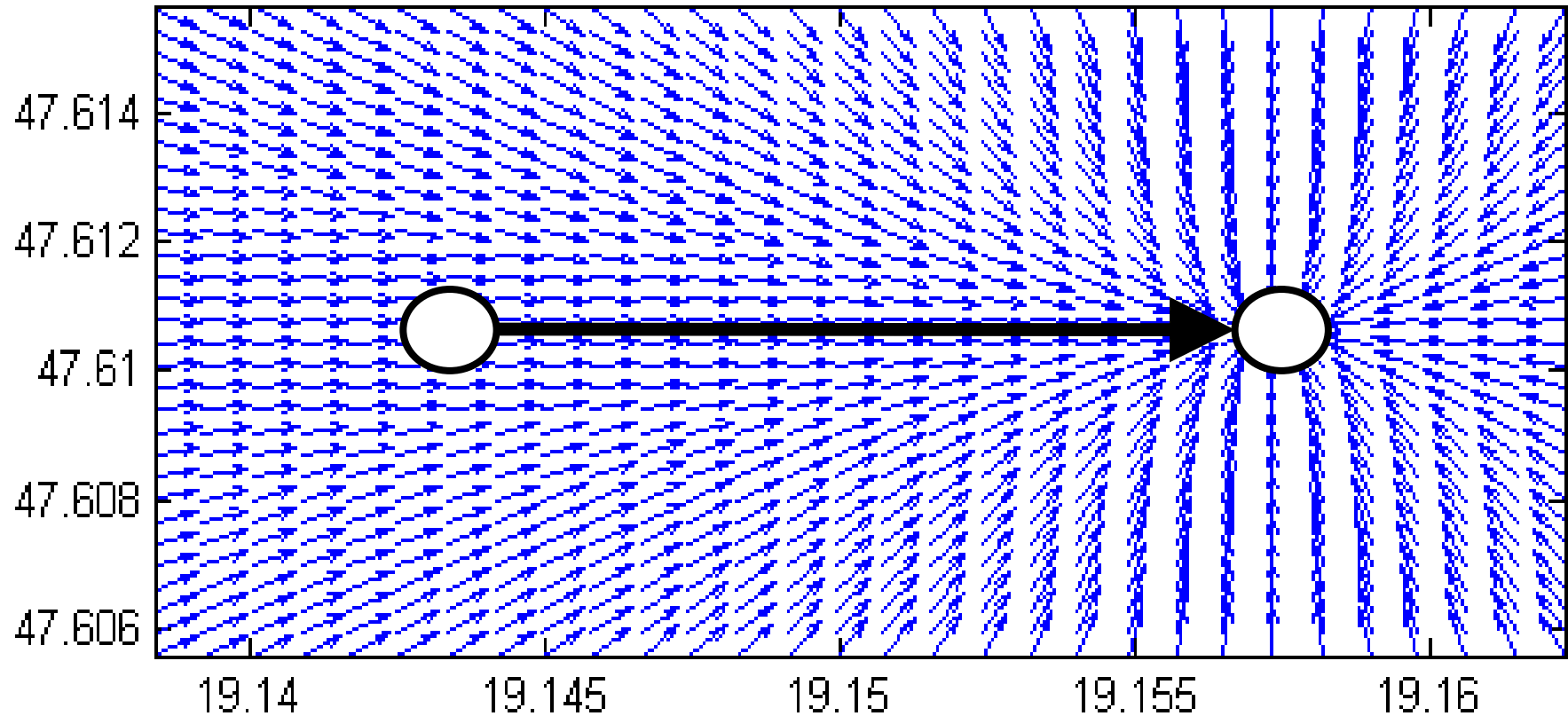
UAV navigáció

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

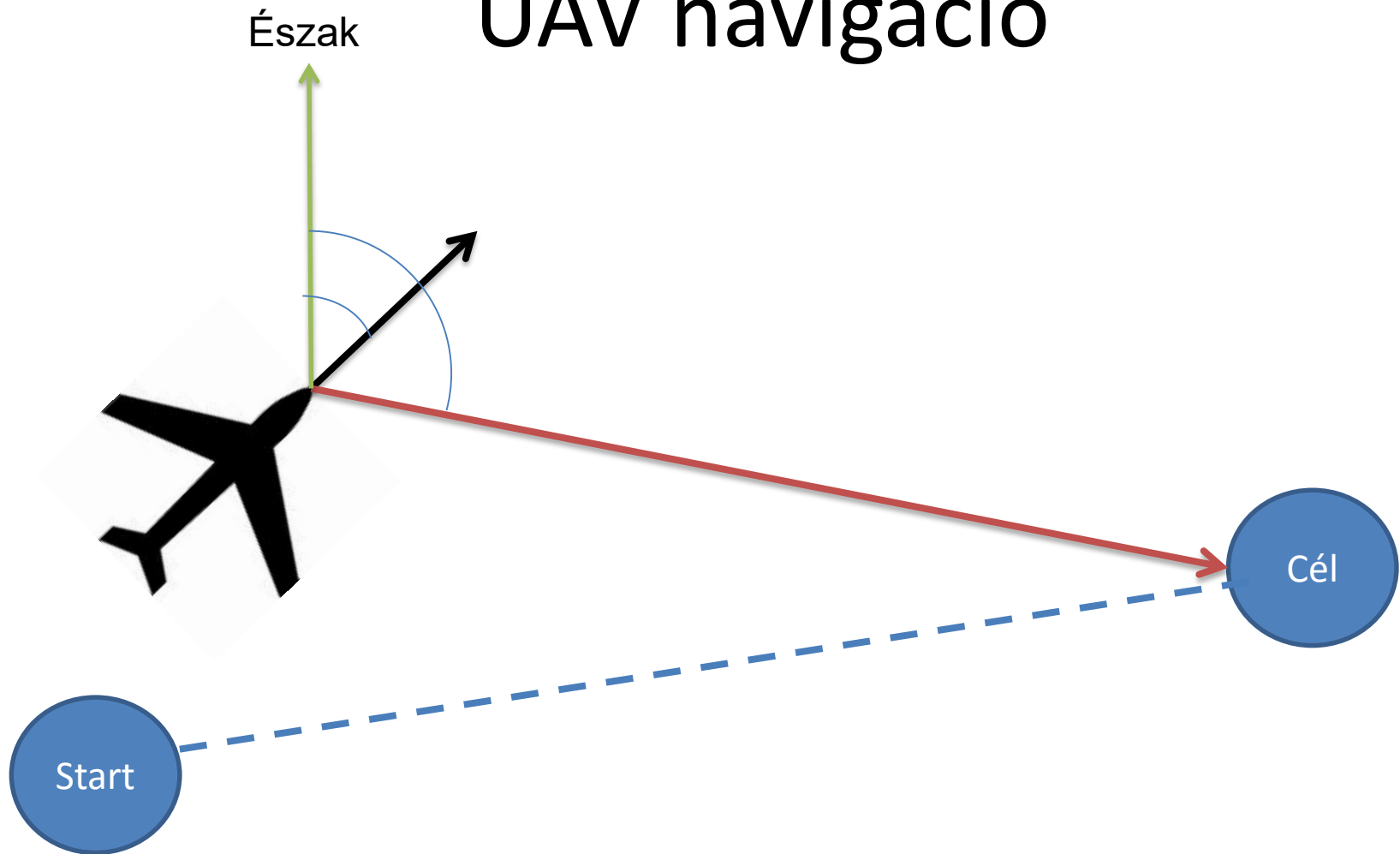


UAV navigáció

- Fordulópontok



UAV navigáció



(Szöghiba: -180° és $+180^\circ$ között!)






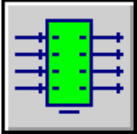





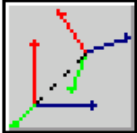


Aerosonde „Laima” UAV



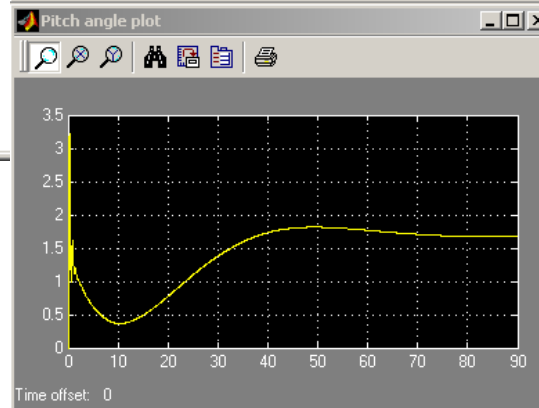
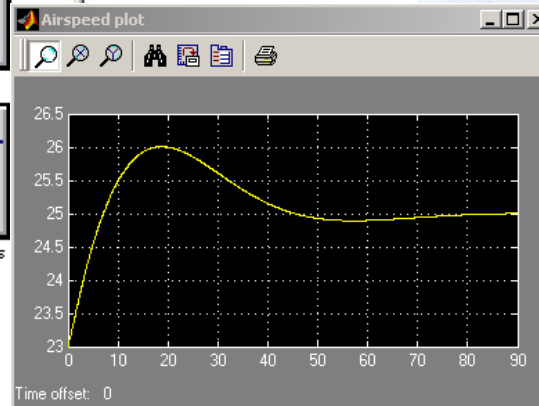
Aerosim

Library: aerosimlib

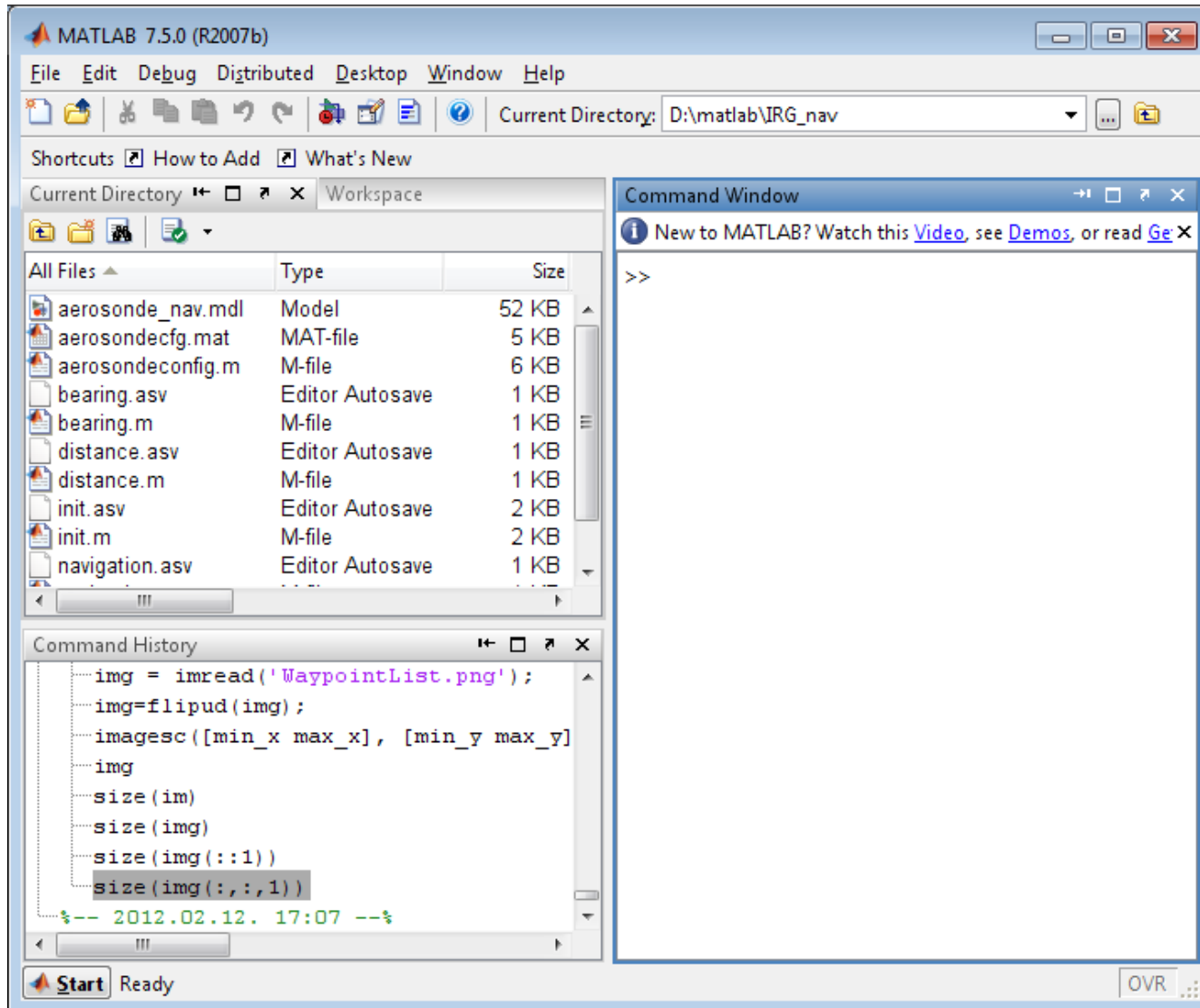
File Edit View Format Help

 Actuators	 Aerodynamics	 Atmosphere	 Earth
 Equations of Motion	 Complete Aircraft	 Inertia	 Math
 Pilot Interface	 Propulsion	 Sensors	 Transformations
 Unit Conversion	 FlightGear- Compatible FDM		

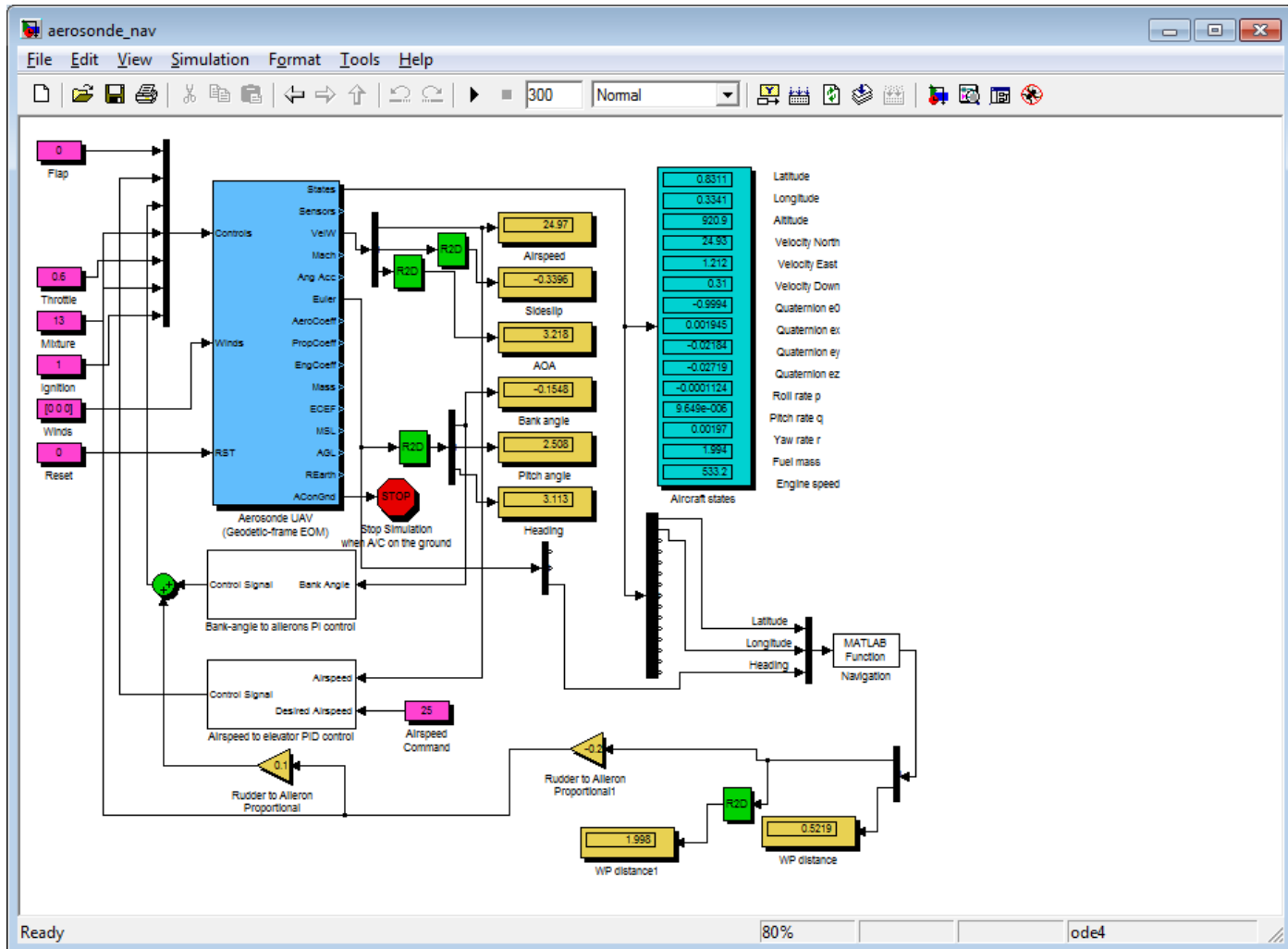
AeroSim Blockset 1.01
Copyright (c) 2002 Unmanned Dynamics, LLC.



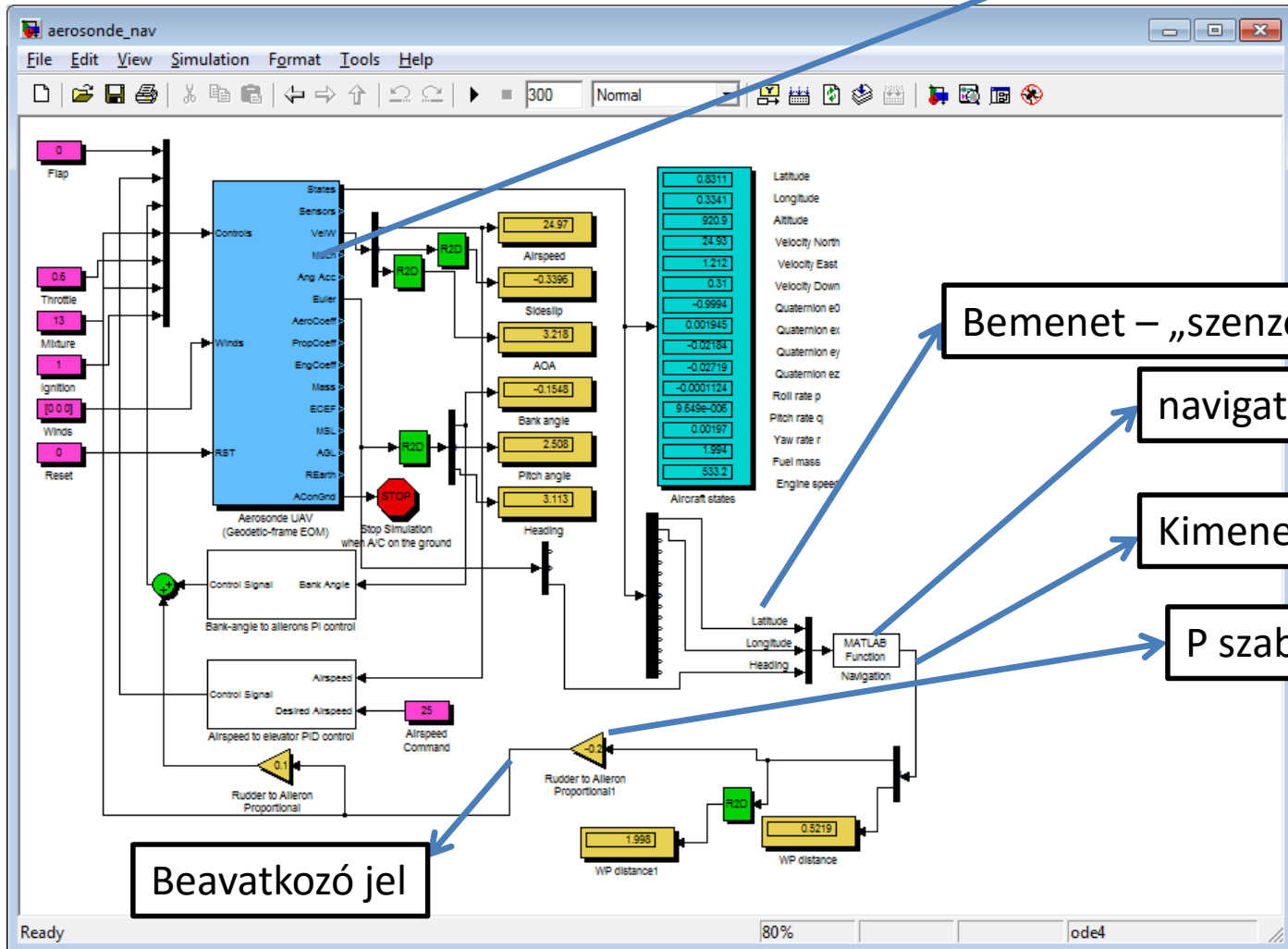
Matlab



Modell



Modell

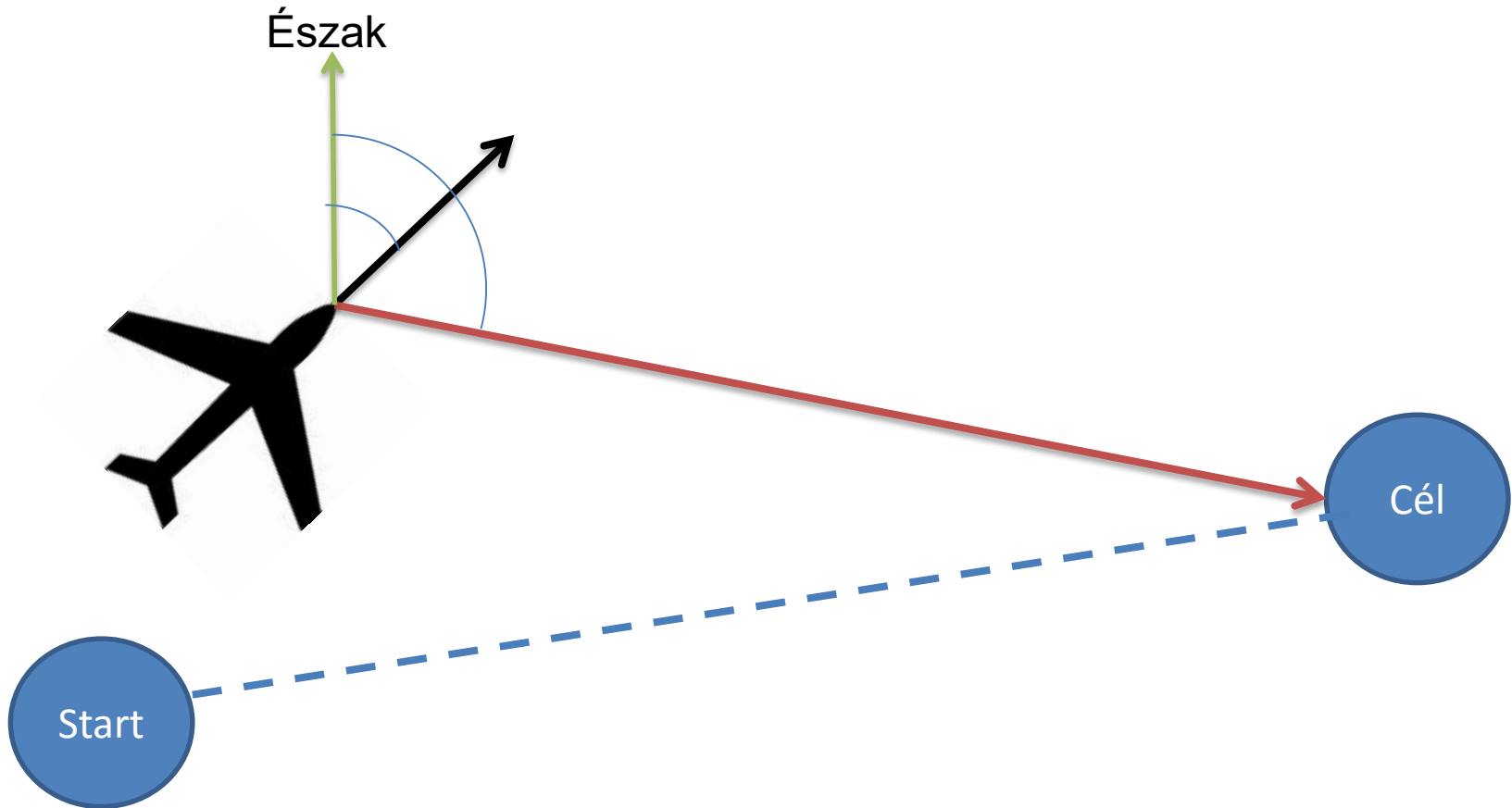


Inicializálás

- Init.m

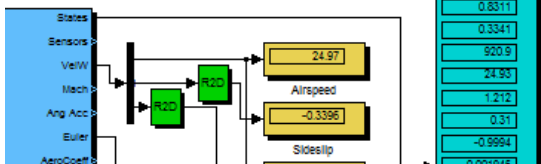
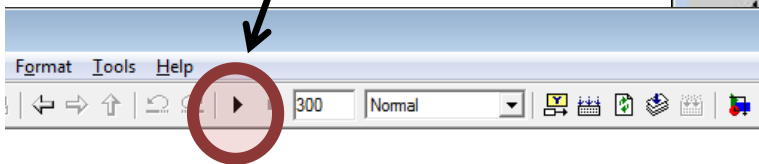
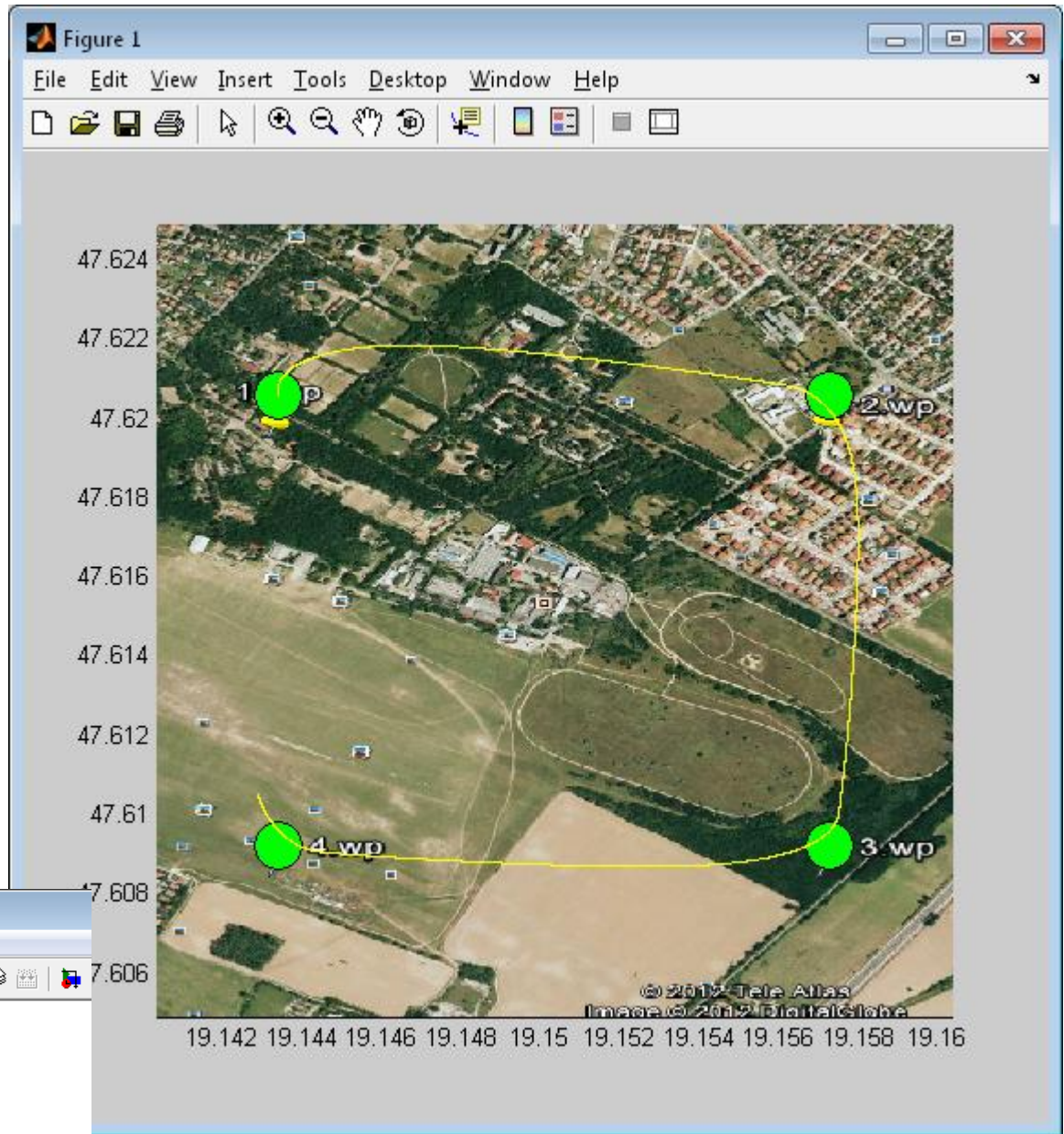
```
1 function [ output_args ] = init( input_args )
2 %INIT Summary of this function goes here
3 % Detailed explanation goes here
4 - clear all;
5 - close all;
6
7 - global WPTC
8 - WPTC=2; %CélWP sorszáma
9
10 - global lat;
11 - global lon;
12 - lat=1;
13 - lon=2;
14
15 %Dunakeszi reptér 4 * (kb)1000m
16 - global WPTList
17 - WPTList=[
18     47.6206, 19.1434; %1. WP fokban
19     47.6206, 19.1573; %2. WP fokban
20     47.6092, 19.1573; %3. WP fokban
21     47.6092, 19.1434 ];%4. WP fokban
22 - WPTList=deg2rad(WPTList)
23 %1-től számoz, nem nullától!!!
24 %WPTList(1,lat) : 47.6206
25 %WPTList(1,lon) : 19.1434
26
27 - plotMap();
28
--
```

(legegyszerűbb) UAV navigáció



(Szöghiba: -180° és $+180^\circ$ között!)

- Futtatás:
 - init.m futtatása (új térkép)
 - modell futtatása

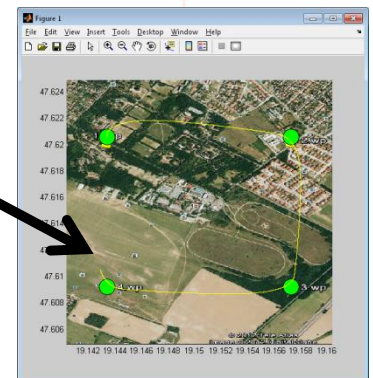
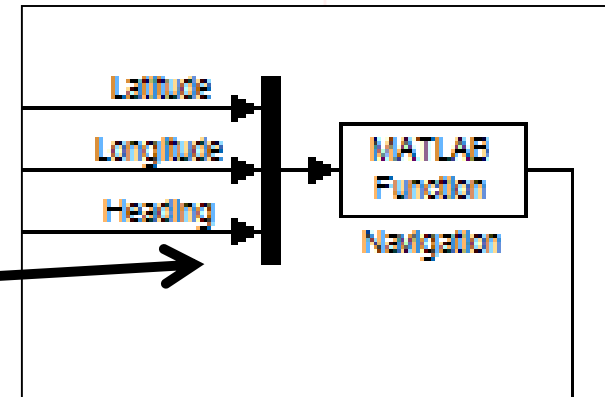


Navigation.m

```
1 function [ output_args ] = navigation( input_args )
2
3 %globális változók behívása
4 - global WPTC; %cél fordulópont sorszáma a listán belül
5 - global lat; % UAV latitude
6 - global lon; % UAV longitude
7 - global WPTList; % fordulópont lista
8 - global h; % plot handler
9 - waypoint_radius=100; %[méter] fordulópont rádiusza
10
11 %input argumentumok 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 WPTC; %cél fordulópont sorszáma a listán belül
5 global lat; % UAV latitude
6 global lon; % UAV longituda
7 global WPTList; % fordulópont lista
8 global h; % plot handler
9 waypoint_radius=100; %[méter] fordulópont rádiusza
10
11 %input argumentumok 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
```



```
23 % 1; adott pontban (UAV) a kívánt haladási irány kiszámítása
24 %bearing.m-et felhasználva -> bearing([TO_lat TO_lon FROM_lat FROM_lon]); tömbként megadva
25 %
26 %
27 %
28 - desiredBearing=0;
29
30 % 2; UAV - cél fordulópont távolság számítása
31 %distance.m-et felhasználva -> distance([TO_lat TO_lon FROM_lat FROM_lon]); tömbként megadva
32 %
33 %
34 %
35 - WPDistance=0;
36
37 % 3; fordulópont léptetése, ha a fordulópont távolsága a fordulópont
38 % rádiusznál kisebb
39 %
40 %
41 %
42 - WPTC=WPTC;
43
44 % 4. szöghiba (kívánt és valós haladási irány különbsége, -Pi...+Pi (-180...+180°) között megadva)
45 %
46 %
47 %
48 %
49 - headingError=0;
50
51 %kimenet
52 - output_args=[headingError, WPDistance/1000] % szöghiba radiánban és távolság méterben
53
```