Intelligent Systems
Lab

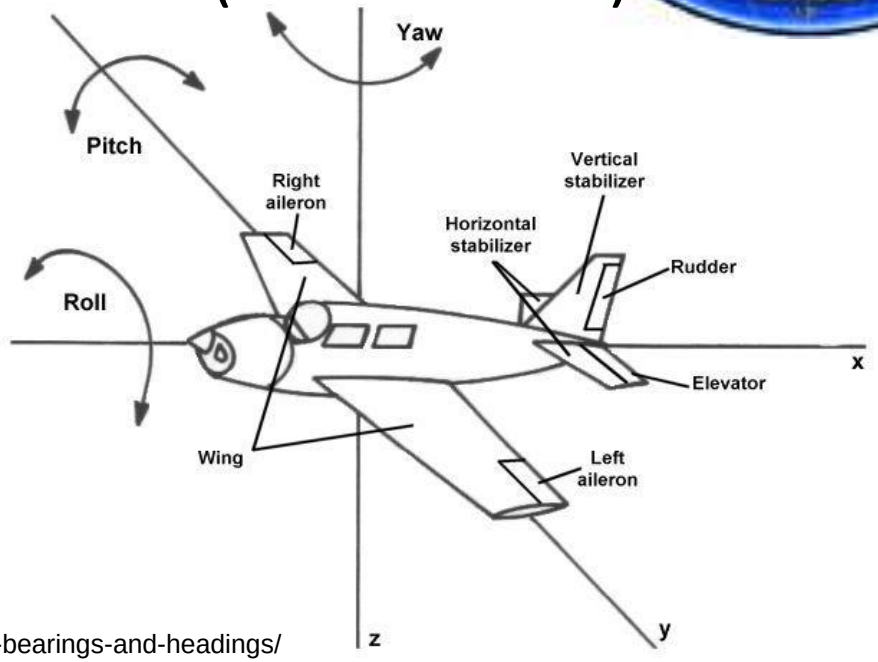
Robots 4.

http://mobil.nik.uni-obuda.hu/en/

UAV

UAV guidance

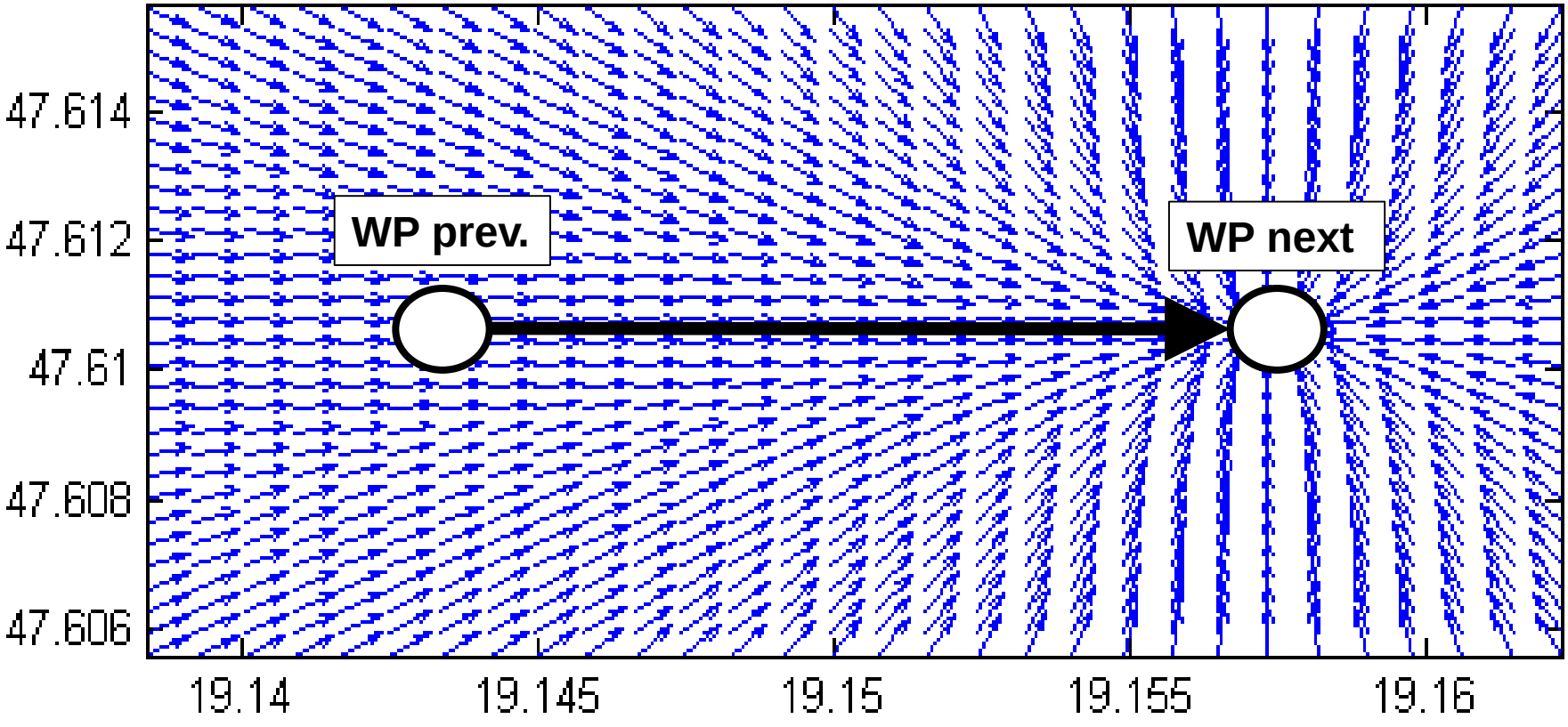
- Course
  - Planned flightpath „direction”
- Bearing
  - Actual flight „direction” (North based)
- Heading
  - Nose „direction”



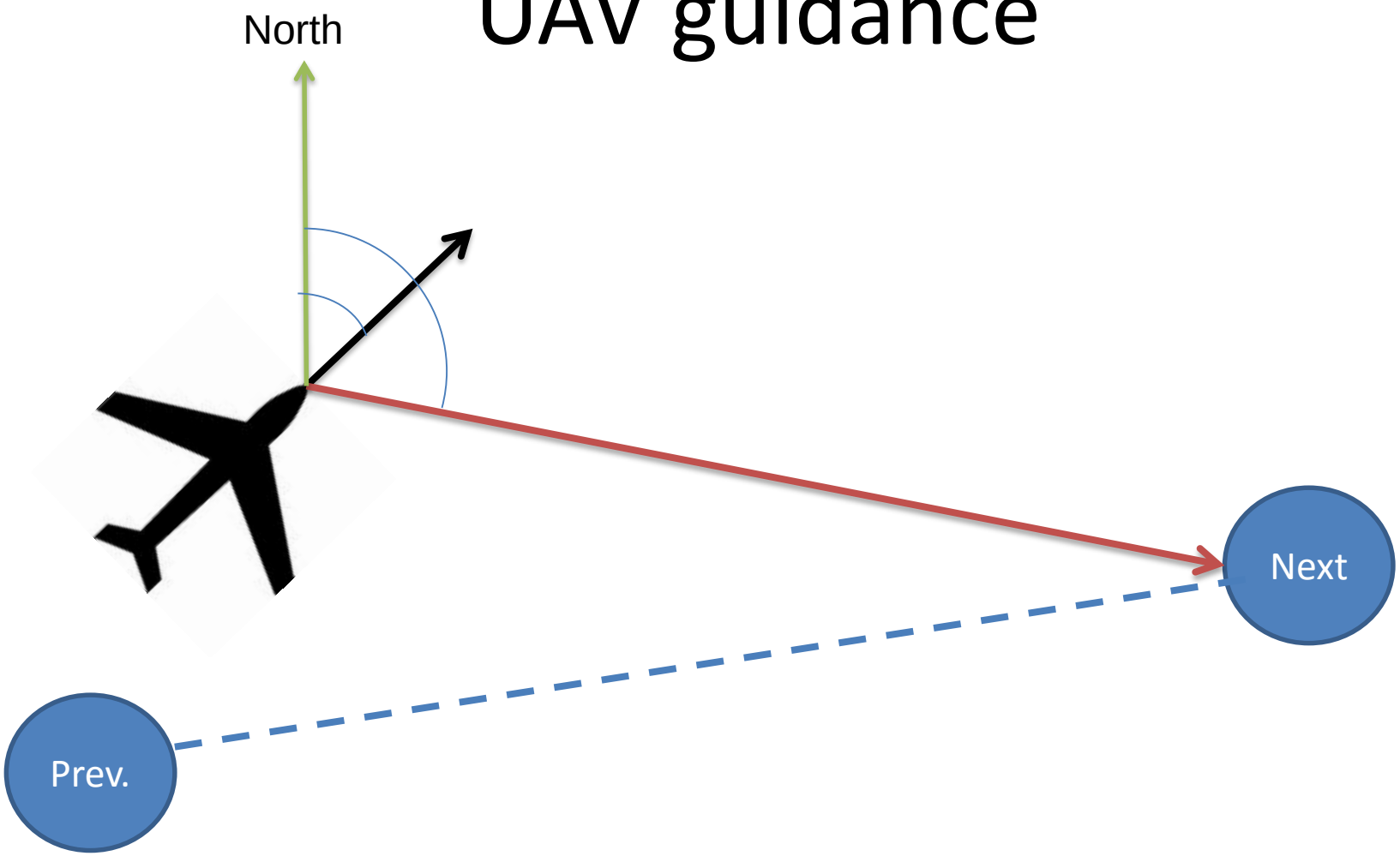
http://www.paddlinglight.com/articles/technique/navigation-course-bearings-and-headings/

UAV guidance

- Based on waypoints



UAV guidance



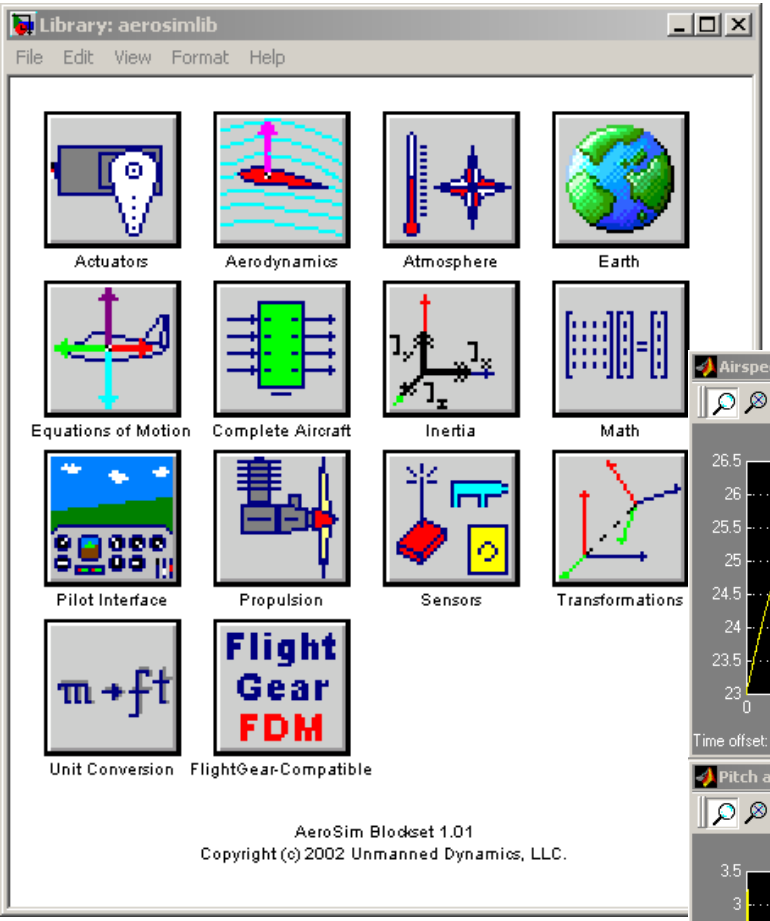
(Angle error: -180° … +180°!)

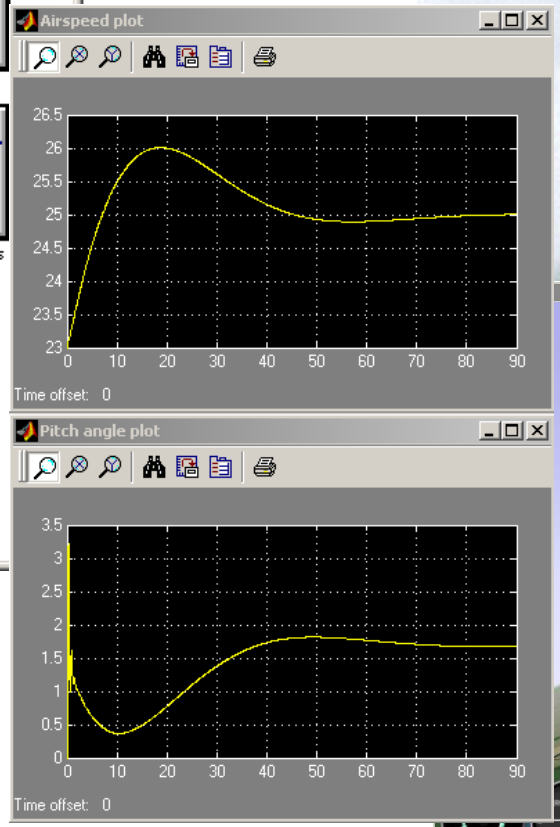
http://www.movable-type.co.uk/scripts/latlong.html

Aerosonde „Laima” UAV

http://www.barnardmicrosystems.com/UAV/milestones/atlantic_crossing_1.html

Aerosim

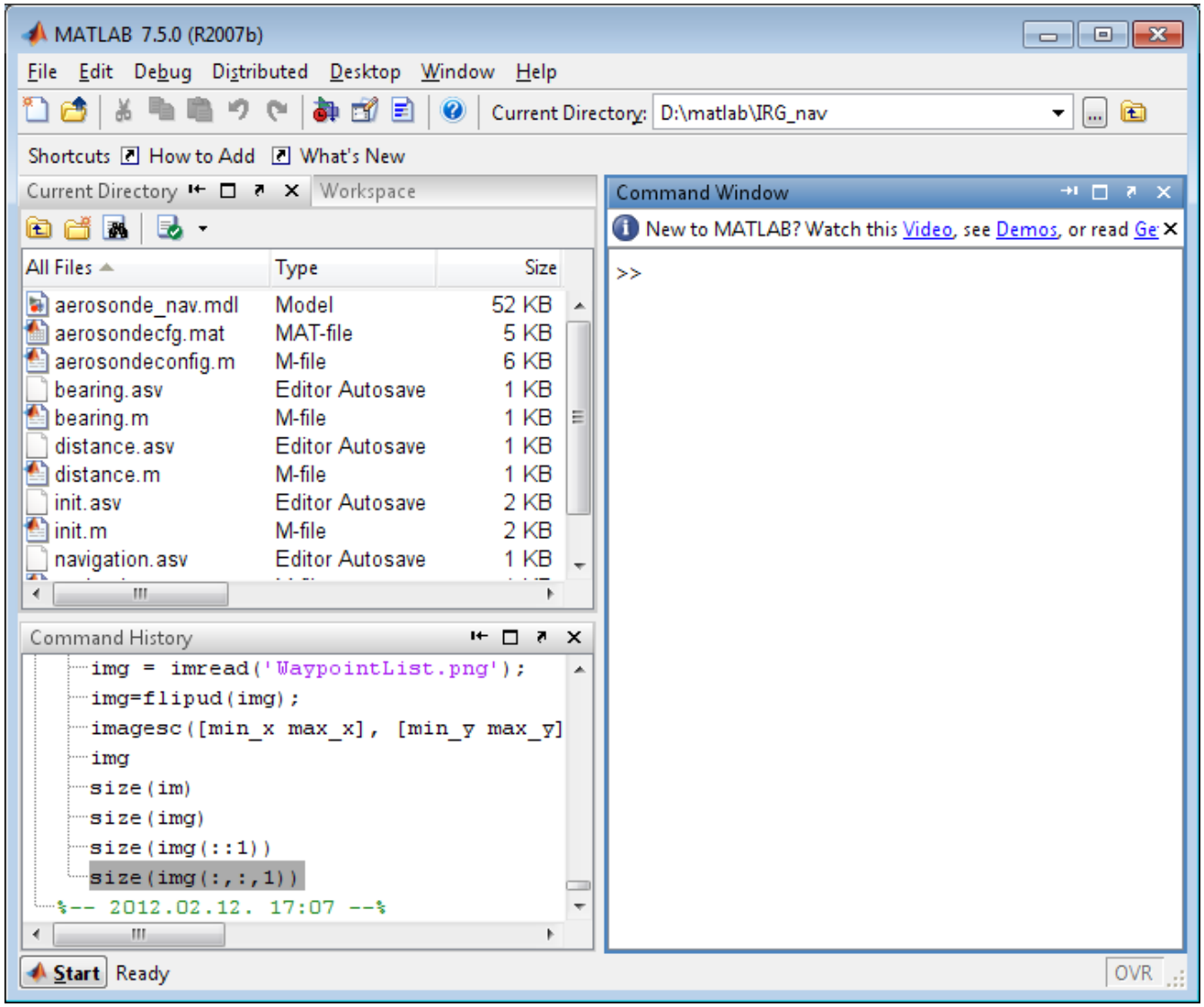

Library: aerosimlib
File Edit View Format Help
Actuators
Aerodynamics
Atmosphere
Earth
Equations of Motion
Complete Aircraft
Inertia
Math
Pilot Interface
Propulsion
Sensors
Transformations
m → ft
Unit Conversion
Flight Gear FDM
FlightGear-Compatible
AeroSim Blockset 1.01
Copyright (c) 2002 Unmanned Dynamics, LLC.


Airspeed plot
26.5
26
25.5
25
24.5
24
23.5
23
0 10 20 30 40 50 60 70 80 90
Time offset: 0
Pitch angle plot
3.5
3
2.5
2
1.5
1
0.5
0
0 10 20 30 40 50 60 70 80 90
Time offset: 0

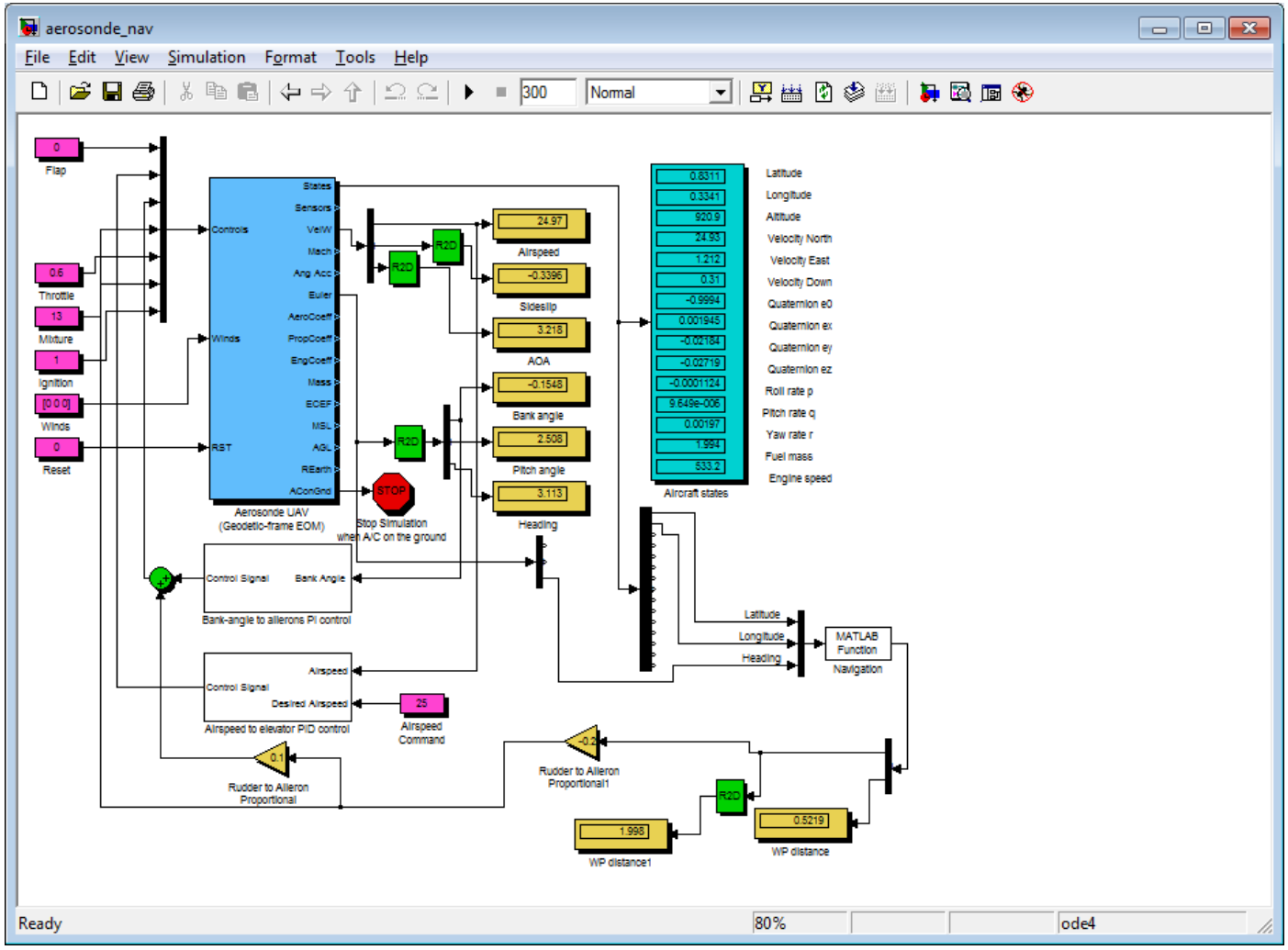

AEROSONDE


BRAKE

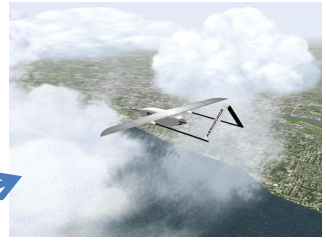
Matlab

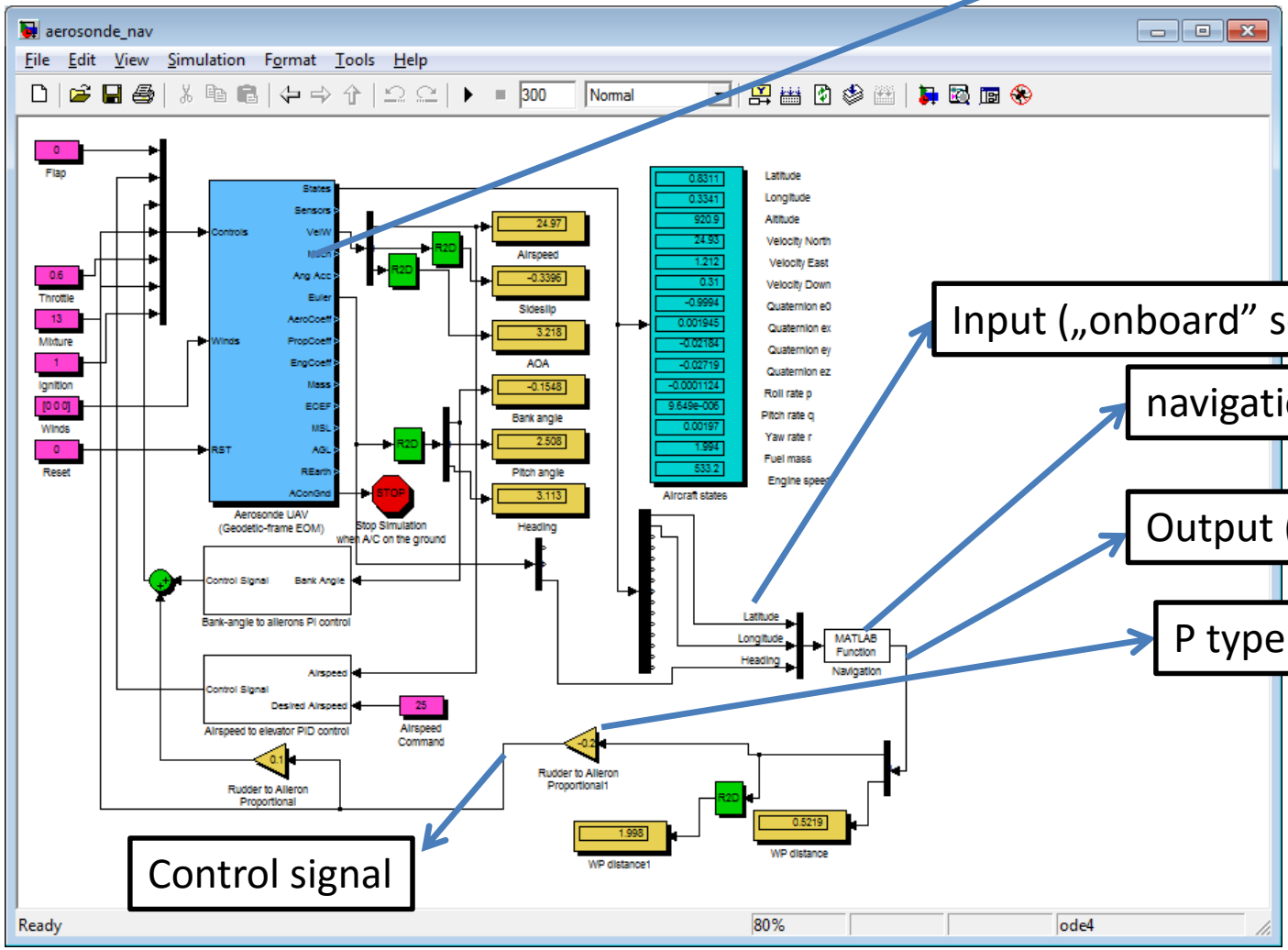

MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: D:\matlab\IRG_nav
Shortcuts How to Add What's New
Current Directory
Workspace
All Files Type Size
aerosonde_nav.mdl Model 52 KB
aerosondecfg.mat MAT-file 5 KB
aerosondeconfig.m M-file 6 KB
bearing.asv Editor Autosave 1 KB
bearing.m M-file 1 KB
distance.asv Editor Autosave 1 KB
distance.m M-file 1 KB
init.asv Editor Autosave 2 KB
init.m M-file 2 KB
navigation.asv Editor Autosave 1 KB
Command Window
New to MATLAB? Watch this Video, see Demos, or read Ge
>>
Command History
img = imread('WaypointList.png');
img=flipud(img);
imagesc([min_x max_x], [min_y max_y]
img
size(im)
size(img)
size(img(::1))
size(img(:,:,1))
%-- 2012.02.12. 17:07 --%
Start Ready
OVR

Modell


aerosonde_nav
File Edit View Simulation Format Tools Help
300
Normal
Flap
Throttle
Mixture
Ignition
Winds
Reset
Controls
Winds
RST
States
Sensors
VelW
Mach
Arg Acc
Euler
AeroCoeff
PropCoeff
EngCoeff
Mass
ECEF
MSL
AGL
REarth
AConGnd
Aerosonde UAV
(Geodetic-frame EOM)
R2D
Airspeed
Sideslip
AOA
Bank angle
Pitch angle
Heading
STOP
Stop Simulation
when A/C on the ground
Aircraft states
Latitude
Longitude
Altitude
Velocity North
Velocity East
Velocity Down
Quaternion e0
Quaternion ex
Quaternion ey
Quaternion ez
Roll rate p
Pitch rate q
Yaw rate r
Fuel mass
Engine speed
Control Signal
Bank Angle
Bank-angle to ailerons PI control
Airspeed
Control Signal
Desired Airspeed
Airspeed to elevator PID control
Airspeed
Command
Rudder to Aileron
Proportional
Rudder to Aileron
Proportional1
MATLAB
Function
Navigation
WP distance1
WP distance
Ready
80%
ode4

Modell





Input („onboard" sensor data)

navigation.m

Output (angle error

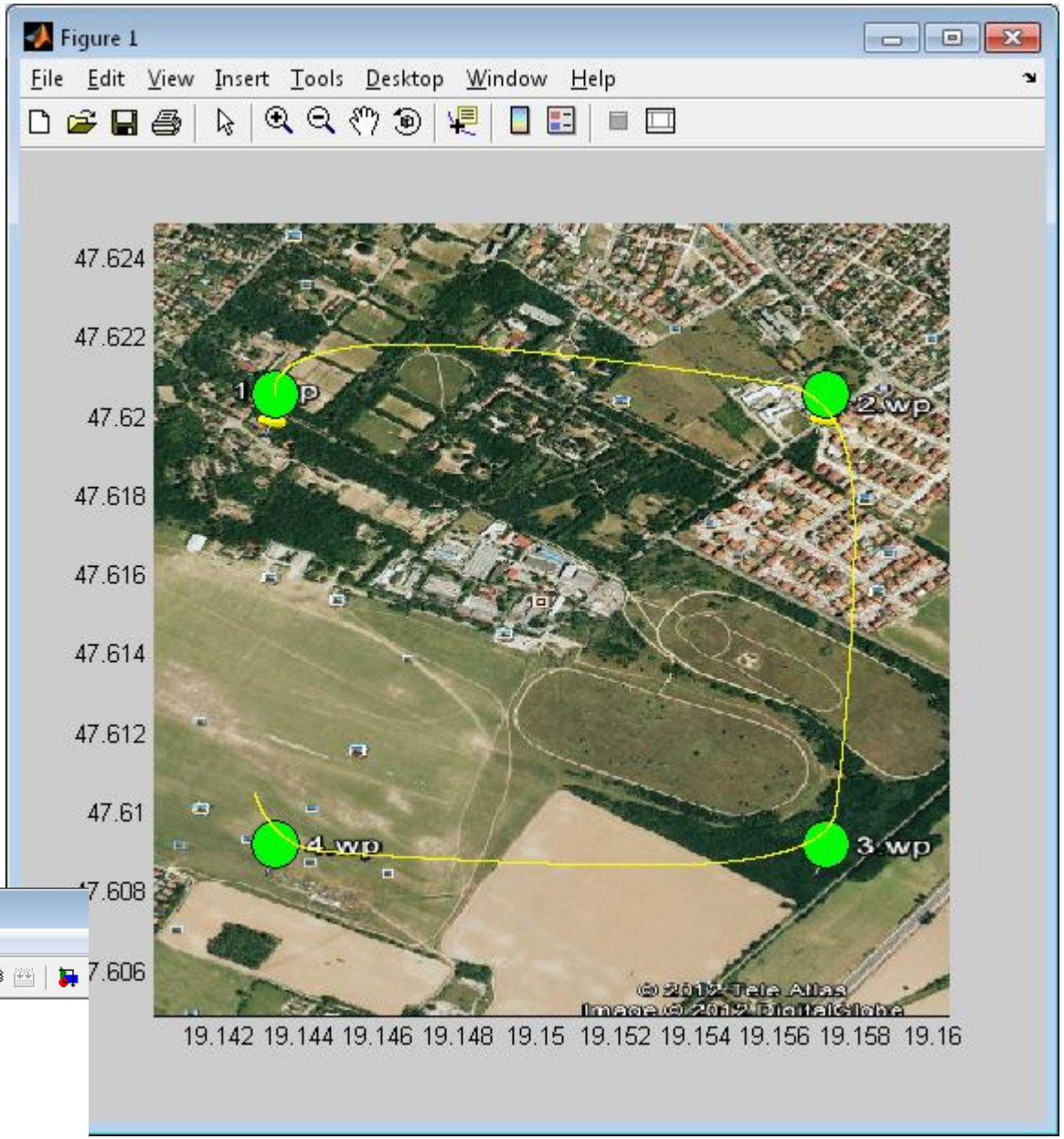
P type controller

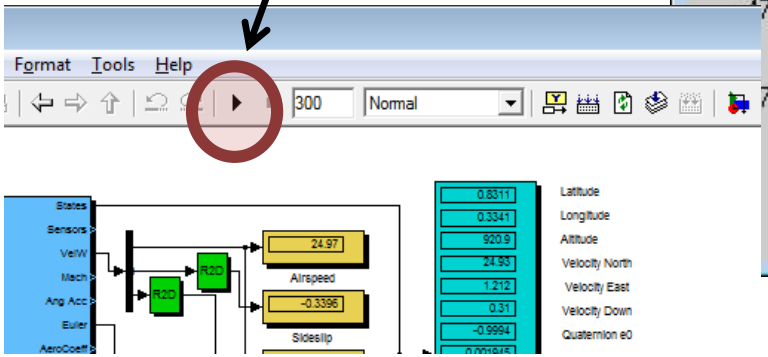
Control signal

init.m

```
function [ output_args ] = init( input_args )
%INIT Summary of this function goes here
%   Detailed explanation goes here
clear all;
close all;

global WPTC
WPTC=2;   %WayPoinT Counter (Next waypoint ID)

global lat;
global lon;
lat=1;
lon=2;

%LHDK (Dunakeszi Airport) 4 * (~)1000m
global WPTList
WPTList=[
    47.6206, 19.1434; %1. Waypoint given in DD (DecimalDegree)
    47.6206, 19.1573; %2. Waypoint given in DD (DecimalDegree)
    47.6092, 19.1573; %3. Waypoint given in DD (DecimalDegree)
    47.6092, 19.1434 ];%4. Waypoint given in DD (DecimalDegree)
WPTList=deg2rad(WPTList); % Matlab computes with RADs instead of DEGs
%Array first element: 1 (NOT 0 based as C/++/# !)
%WPTList(1,lat) : 47.6206
%WPTList(1,lon) : 19.1434

plotMap();

```

- How to run:
  - Run init.m every time! (clears map& vars)
  - Run model simulation



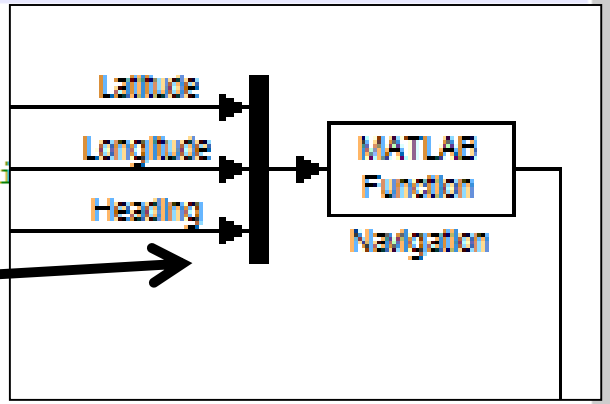


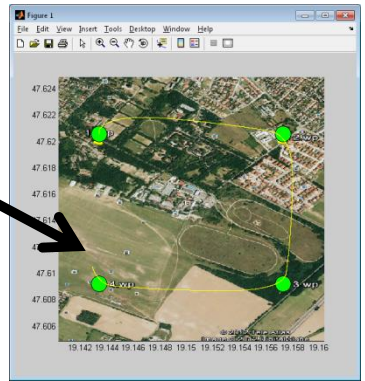
Navigation.m

```
function [ output_args ] = navigation( input_args )

% GLOBAL DEFINE: HEADING now equals BEARING, slideship angle = 0!

%include global vars
global WPTC; %WayPoinT Counter (Next waypoint ID)
global lat; % UAV latitude
global lon; % UAV longitude
global WPTList; % WayPoinT list
global h; % plot handler
waypoint_radius=100; %[meter] waypoint radius (e radius circle)

%set UAV inputs
UAV_lat=input_args(1);
UAV_lon=input_args(2);
UAV_heading=input_args(3);

hold on; %holding old plot values (prev. positions)
plot(rad2deg(UAV_lon), rad2deg(UAV_lat),'--y'); % plot new position

```

Navigation.m

```
1   function [ output_args ] = navigation( input_args )
2
3   % GLOBAL DEFINE: HEADING now equals BEARING, slideship angle = 0!
4
5   %include global vars
6   global WPTC; %WayPoinT Counter (Next waypoint ID)
7   global lat; % UAV latitude
8   global lon; % UAV longitude
9   global WPTList; % WayPoinT list
10  global h; % plot handler
11  waypoint_radius=100; %[meter] waypoint radius (e radius ci
12
13  %set UAV inputs
14  UAV_lat=input_args(1);
15  UAV_lon=input_args(2);
16  UAV_heading=input_args(3);
17
18  hold on; %holding old plot values (prev. positions)
19  plot(rad2deg(UAV_lon), rad2deg(UAV_lat),'--y'); % plot new position
20
```





```
%HELP
%http://www.mathworks.com/help/techdoc/


% 1; calculate the current bearing from the current position (UAV_lat/UAV_lon)
% using bearing.m -> bearing([TO_lat TO_lon FROM_lat FROM_lon ]); input is a matlab
array
%
%
%
desiredBearing=0;

% 2; calculate UAV - Newt WPT distance
% using distance.m -> bearing([TO_lat TO_lon FROM_lat FROM_lon ]); input is a matlab
array
%
%
%
WPDistance=0;

% 3; WayPoinT Counter (WPTC) incrementation (0...3) if actual UAV-WPT distance is
less than waypoint_radius
%
%
%
%
WPTC=WPTC;

% 4. Calculate the angle error (difference of desiredBearing and UAV_heading)
% output must be between -Pi and Pi (-180...180°)
%
%
%
%
headingError=0;

%return output
output_args=[headingError, WPDistance/1000]; % angle error [rad], Waypoint distance
[m]

```