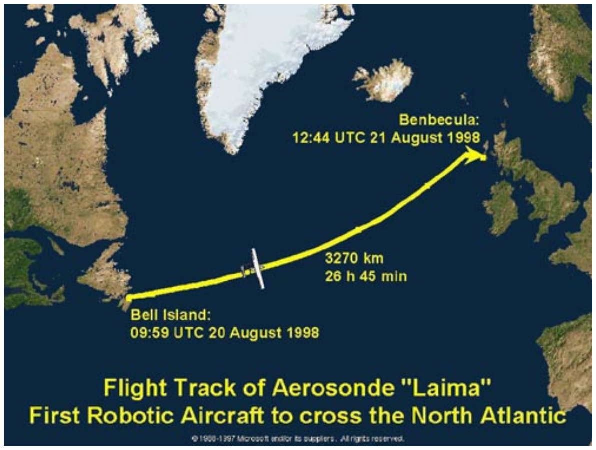
# Final Project for

## EE392m - Control Engineering in Industry, Spring 2005

### Problem description and background

The problem is to design a guidance system for the Aerosonde UAV (Unmanned Air Vehicle), see the picture below. This Australian UAV was the first one to cross over the Atlantic in a completely autonomous flight.



The guidance system design in this project will build upon Matlab Aerosim blockset, where a full simulation model of the Aerosonde UAV is implemented and documented and the initial design of the autopilot provided at the class website. As a first step in the project development you will need to download and install these simulations and be able to run them.

- Download Aerosim blockset and documentation from http://www.u-dynamics.com/aerosim/; free for academic use. Install Aerosim blockset in the default directory.
- Make sure you can run Aerosonde simulation demos provided with the blockset.
- Download the Simulink file `Aerosonde_FinalEE392m.mdl` and Matlab S-function `GuideEE392m.m` and place them into `./Aerosim/samples` folder along with other Aerosonde simulations.
- The file AIAA2001-016.pdf on the class website describes a simple lateral guidance algorithm by the Aerosim blockset developer. This algorithm is implemented in `GuideEE392m.m` as a baseline reference for performance of your guidance algorithm.

### Final project work

The project consist of several consecutive problems and solution of each next problem builds upon the solution of the previous one. The project report should contain a section for each problem describing (i) the solution approach and any necessary justification (ii) the designed algorithm software (Simulink diagram or Matlab code) and (iii) simulation results demonstrating performance of the algorithm in solving the problem.

The overall goal, to which all of the problems contribute, is to design a guidance system that will ensure that the UAV tracks a straight line 3-D trajectory characterized by a constant altitude of 900m, waypoint (N-E coordinates), and a heading angle for the straight line passing though the waypoint. As a minimum, the tracking should be demonstrated in the absence of the disturbances with convergence from an off-track initial position. Ideally, you should demonstrate a reasonable quality of tracking despite the winds (as simulated in the standard Aerosim model).

In addition to handing in the project paper, you will be required to make a short 5 min presentation (2-3 slides). It should summarize the features and illustrate the performance of the designed system.

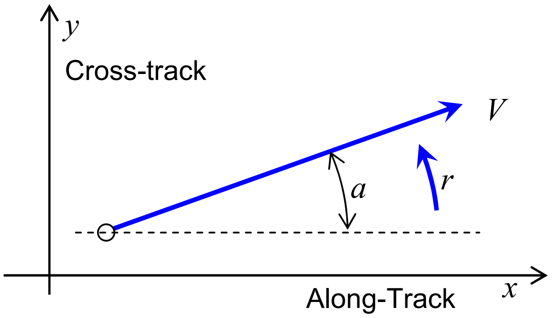
## Problem 1. Design an altitude hold controller and add it to the autopilot.

The Autopilot block in the simulation `Aerosonde_FinalEE392m.mdl` currently contains the lateral control logic that provides tracking of the given yawing rate (turn rate) and the longitudinal control logic that provides tracking of the pre-set pitch angle.

Extend the autopilot by designing controller for holding the commanded altitude. This could be a cascade PI controller providing pitch command to the pitch angle control loop.

## Problem 2. Design a state-feedback MPC controller, demonstrate in a simple sampled time simulation

Following a straight line trajectory might be needed for aerial surveillance of a road (see the picture). Consider the following simple sampled-time model for the cross-track guidance of the UAV



$$a(t+1) = a(t) + r(t)T_s$$
$$y(t+1) = y(t) + a(t)VT_s + r(t)\cdot 0.5VT_s^2, \qquad (*)$$

where $V$ is the UAV speed, $a$ is the heading angle, $T_s$ is the control sample time, and $y$ is the cross-track coordinate. This simple kinematical model assumes that $V = \text{const}$ and that $a \ll 1$ such that $\sin(a) \approx a$. The states $a$ and $y$ of the system are available at each sample time.

Design a receding horizon controller by solving the following QP optimization problem and applying the first computed control value at each step.

$$J = \|Y\|^2 + r\|DU\|^2 \rightarrow \min$$
$$\text{subject to: } |U| \leq 0.1$$

where $U = [\, r(t+1) \ldots r(t+\mathrm{N})\,]^T$ is the future control sequence vector, $D$ is the first difference matrix, $r$ is a scalar penalty (tuning) weight, $N$ is a prediction horizon and $Y = [\, y(t+1) \ldots y(t+\mathrm{N})\,]^T$ is the predicted output of the system corresponding to $U$ and current state $x(t) = [\, a(t) \;\; y(t)]^T$. All norms are Euclidean 2-norms, the inequality is component wise. The vector $Y$ can be presented in the form

$$Y = Hx + GU,$$

where the entries of the matrices $H$ and $G$ are the initial condition and impulse responses of the model (*).

Implement the controller using the Matlab QUADPROG solver. Validate the controller performance (initial condition response) in a sampled time simulation with the model (*).

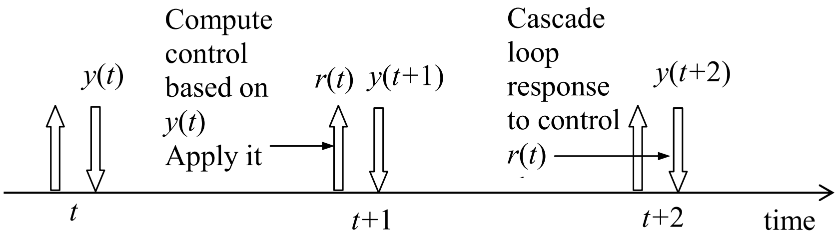
**Problem 3. Design an MPC controller with a sampled time delay, demonstrate in full simulation**

Extend the model (*) to include the sampling time delay. Design an MPC controller using the same approach as outlined in Problem 2. Implement the designed controller in the S-function `GuideEE392m.m` in the place of the baseline cross-track controller. Validate the controller performance in the comprehensive simulation `Aerosonde_FinalEE392m.mdl.` Include the attitude hold controller from Problem 1 in the simulation.

The extended sampled time model taking into account computational delay in the system has the form

$$\begin{aligned} u(t+1) &= r(t) \\ a(t+1) &= a(t) + u(t)T_s \\ y(t+1) &= y(t) + a(t)VT_s + u(t)\cdot 0.5VT_s^2 \end{aligned} \quad , \tag{**}$$

The timeline of the control computations is illustrated in figure below. This timeline shows that the state space model (**) below should be used in lieu of the model (*). It also shows that MPC computations require to store two last control values at each update cycle.



**Problem 4. Add soft constraint for the heading angle.**

The controller designed in Problem 3 has a fundamental issue. It works only for sufficiently small heading angles, e.g., for $|a| < 0.75$ rad. If heading angle becomes larger in the transient, the controller breaks down and UAV gets into a sustained spin. To overcome this issue, design the controller based on the following optimization problem statement

$$\begin{aligned} & J = \|Y\|^2 + q\|W\|_1 + r\|DU\|^2 \to \min \\ & \text{subject to: } |U| \le 0.1, \qquad |A| \le W, \qquad 0.75 \le W \end{aligned} \tag{***}$$

where $U = [\,r(t+1)\ \dots\ r(t+\mathrm{N})\,]^{\mathrm{T}}$ is the future control sequence vector, $D$ is the first difference matrix, $r$ and $q$ are a scalar penalty (tuning) weights, $N$ is a prediction horizon, $Y = [\,y(t+1)\ \dots\ y(t+\mathrm{N})\,]^{\mathrm{T}}$ is the predicted output of the system (**) corresponding to $U$ and current state $x(t) = [\,a(t)\ \ y(t)]^{\mathrm{T}}$, and $A = [\,a(t+1)\ \dots a(t+\mathrm{N})]^{\mathrm{T}}$ is the respective predicted heading angle sequence. The norms in (***) are the Euclidean and 1-norm (sum of absolute values) respectively. All inequalities are component wise. The $N$-component vector $W$ defines an exceedance of the heading angle bound. By making $q$ sufficiently large, this exceedance will be heavily penalized and avoiding it is made a priority for the control. If the heading angle $|a| < 0.75$ at all times, the additional penalty term in the performance index is constant and does not influence the solution.

The problem (***) is a QP problem with the decision vector $[U^{\mathrm{T}}\ \ W^{\mathrm{T}}]^{\mathrm{T}}$.

**Problem 5. (Bonus) Add disturbance compensation.**

The controller designed in Problem 4 has a performance deficiency. The simulation with cross wind shows that there is a substantial steady-state error of tracking caused by the cross-wind disturbance.

Design a disturbance estimator as described in Lecture 15 to ensure an integrator compensation of the steady-state tracking error in the controller. Demonstrate the performance in the simulation with the wind [15 -5 0] (cross wind of 5/ms and tail wind of 15 m/s). Setting up the wind parameters is foreseen in the sim.

*Hint*: the disturbance should enter the model as a cross-track speed offset, not as a position offset.