

MODELLING OF FLIGHT OF AN AUTONOMOUS PARACHUTE

H. Havliš, T. Vyhlídal

Abstract

The paper describes Matlab/Simulink model of flight of an autonomous parachute. It describes the model structure from the equations of motion and their implementation and solution to the visualization of flight using the Matlab Virtual Reality Toolbox.

1 Purpose and basic description of model

The model of flight of an autonomous parachute was constructed in order to study the parachute flight and design of flight automatic control system. By the construction the authors exploited their experiences with modelling and control of flight of an airship and an airplane. There are also several practical results and outputs, based on these experiences, as autopilot structures for unmanned airship (www.airshipclub.com) and ultralight airplane. The core of the model is mathematical description of flight with additional input and output structures (Simulink schemes, m-files) with I/O interfaces as data vectors, GUIs and 3D virtual reality model.

2 Mathematical model of flight

The very basic model of flying object (rigid body with 6 degrees of freedom) is derived and presented for example in [1], [2].

$$\begin{aligned}
 \dot{u} &= r.v - q.w - g.\sin\Theta + \frac{F_x}{m} \\
 \dot{v} &= p.w - r.u + g.\cos\Theta.\sin\Phi + \frac{F_y}{m} \\
 \dot{w} &= q.u - p.v + g.\cos\Theta.\cos\Phi + \frac{F_z}{m} \\
 \dot{p} &= \frac{r.q.(I_z - I_y)}{I_x} + \frac{M_x}{I_x} \\
 \dot{q} &= \frac{r.p.(I_z - I_x)}{I_y} + \frac{M_y}{I_y} \\
 \dot{r} &= \frac{p.q.(I_x - I_y)}{I_z} + \frac{M_z}{I_z} \\
 \dot{\Theta} &= C_\Phi.q - S_\Phi.r \\
 \dot{\Phi} &= p + S_\Phi \tan\Theta.q + C_\Phi \tan\Theta.r \\
 \dot{\Psi} &= S_\Phi \sec\Theta.q + C_\Phi \sec\Theta.r \\
 \dot{x} &= C_\Theta C_\Psi u + (S_\Phi S_\Theta C_\Psi - C_\Phi S_\Psi)v + (C_\Phi S_\Theta C_\Psi + S_\Phi S_\Psi)w \\
 \dot{y} &= C_\Theta S_\Psi u + (S_\Phi S_\Theta S_\Psi + C_\Phi C_\Psi)v + (C_\Phi S_\Theta S_\Psi - S_\Phi C_\Psi)w \\
 \dot{z} &= -S_\Theta u + S_\Phi C_\Theta v + C_\Phi C_\Theta w
 \end{aligned} \tag{1}$$

The sense of the terms of equations is following: g – gravity measuring, $I_{x, y, z}$ – moments of inertia according to aircraft axes, u, v, w velocities according to aircraft axes, p, q, r angular velocities according to aircraft axes, x, y, z – aircraft position (ground coordinates), Θ, Φ, Ψ aircraft Euler angles (pitch, roll and yaw respectively), the notation S_Θ means $\sin(\Theta)$, C_Φ means $\cos(\Phi)$ etc.

There are 12 nonlinear first order differential equations. The first six equations describe the flight dynamics (force equations, momentum equations), the last six equations describe the kinematics of the motion – transformation between aircraft and ground axes system. The terms F_x , F_y , F_z , M_x , M_y , M_z mean the outer forces and moments according to the aircraft axes x , y , z respectively. There are aerodynamical actions (buoyancy, drag), engine actions, actuator actions etc. Note that by changing of the outer forces and moments it is easy to modify the model by that way, that it can describe the flight of airplane, airship, parachute or any other flying object. For example (very simply) the airspeed dependent buoyancy describes the movement of an airplane, constant buoyancy describes the movement of airship, no engine forces respond the parachute model etc.

3 Matlab/Simulink model implementation, model features

The heart of simulation model is Simulink structure, resolving the above mentioned equations system (1), see Figure 1.

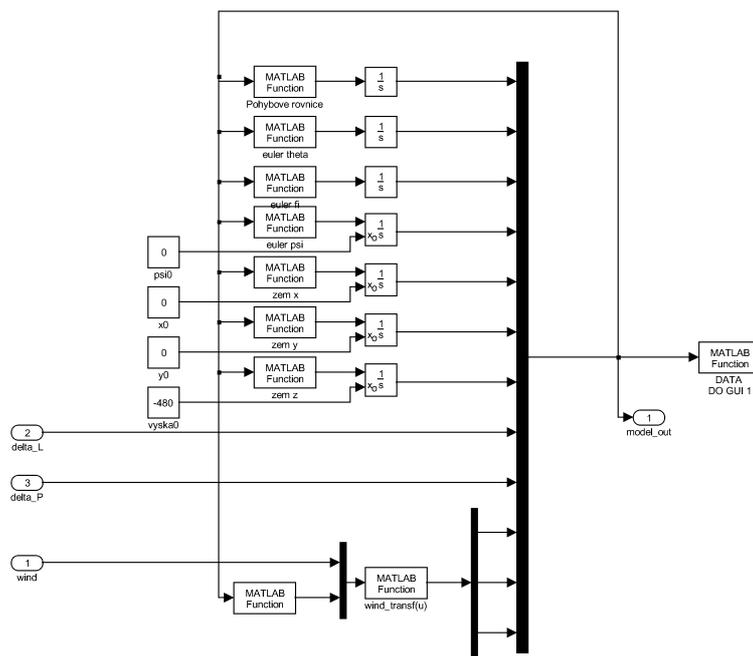


Figure 1. Heart of the model, Simulink scheme resolving the equations system (1)

The scheme shows us the basic principle of solving of the equations system (1). There are seven integrator blocks, first six equations (force and moment) are computed simultaneously in one block as 6x1 vector in order to simplify the structure. The system enables setting of some of initial conditions. The initial conditions that can be set are initial yaw, position and altitude; Another initial conditions (pitch and roll angles and velocities) are not supposed to be important and are equal to zero, because the steady-state flight appears several seconds after start of the simulation when there are no disturbances and actuators influence. The initial conditions can be set before starting of the simulation from GUI. The computation runs over 17x1 data vector, which includes 12 state variables, 2 actuating signals (2 flaps) and 3 error signals (wind velocity in 3 axes).

As error (disturbance) input is considered the wind, defined as a 3x1 vector (three directions, magnitude). The model enables to set the wind velocity in ground coordinate system manually or use special block with pre-defined altitude dependent wind profile for any number of points (altitudes). The proper wind direction/velocity between these points is being computed as a linear interpolation.

The model enables testing of algorithms for computing or guessing of the wind velocity and direction. This task is in the parachute flight control very important, because some landing points can be by strong wind inaccessible. The final part of the landing maneuver is also necessary to be done in the opposite direction of wind flow. The algorithms are based on the principles of unknown-input-estimation, state observers etc.

The model has plenty of physical parameters, for example parachute mass, dimensions, air density, moments of inertia, surfaces, aerodynamical coefficients (drag, buoyancy...) etc. These physical parameters are designed as Matlab global variables and are concentrated in a ini m-file, that must be processed before the start of simulation.

For comfortable launch and control of the model was created Matlab GUI with the basic parameters, initial conditions and control gauges (Fig. 2).

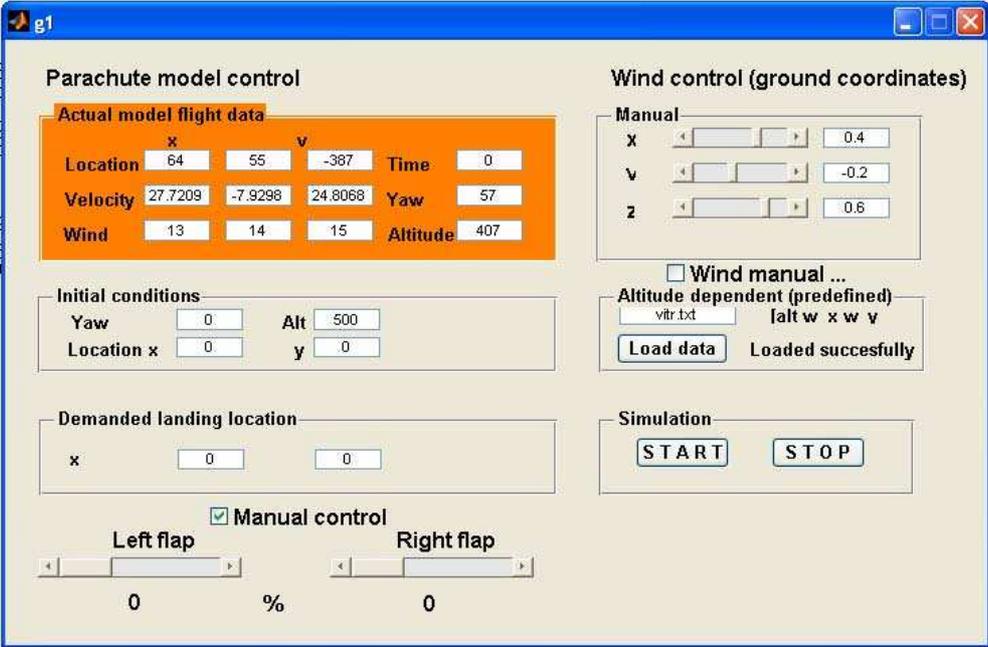


Figure 2. Gui for basic control of the parachute model

The purpose of model design is the automatic flight control system design, the authors constructed several types/architectures of the control system. The control system has some control levels, similar as in [3], from the basic levels as control of the Euler angles, over the straight line flight control and finally the trajectory planning according to demanded landing location, its position, wind influence etc.

Final version of control system will be implemented in the AMPAD projects (automatic cargo parachute), resolved in collaboration of the Centre of Applied Cybernetics and companies Kybertec and TL Electronic. The project is managed and owned by the TL Electronic.

All flight parameters, state variables, actuators movement, real wind direction, estimated wind direction etc. can be logged as data vectors for later proceeding. The record of manually controlled flight of parachute (its position in ground coordinates) shows Figure 3.

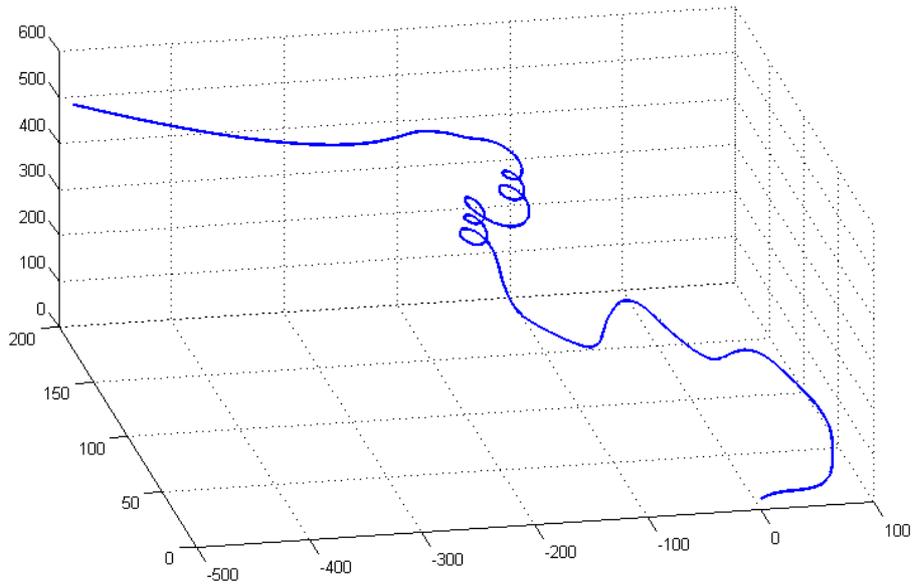


Figure 3. Example of record of a trajectory by manually controlled flight of parachute (initial altitude 500 m)

4 Virtual reality visualisation of parachute flight

The 3D virtual reality model was created using the VRealm builder, which is included in the Matlab/Simulink software. Several model outputs, e. g. the position of parachute and its Euler angles, are sent via the VRSink block (Virtual reality toolbox). The flight is being visualized by the translations and rotations of the parachute in the space and the actuators (control flaps) position is also controlled.

There are many “background” additional objects (mountains, buildings etc.) included into the 3D model from the libraries of Vrealm builder, see Fig. 4.

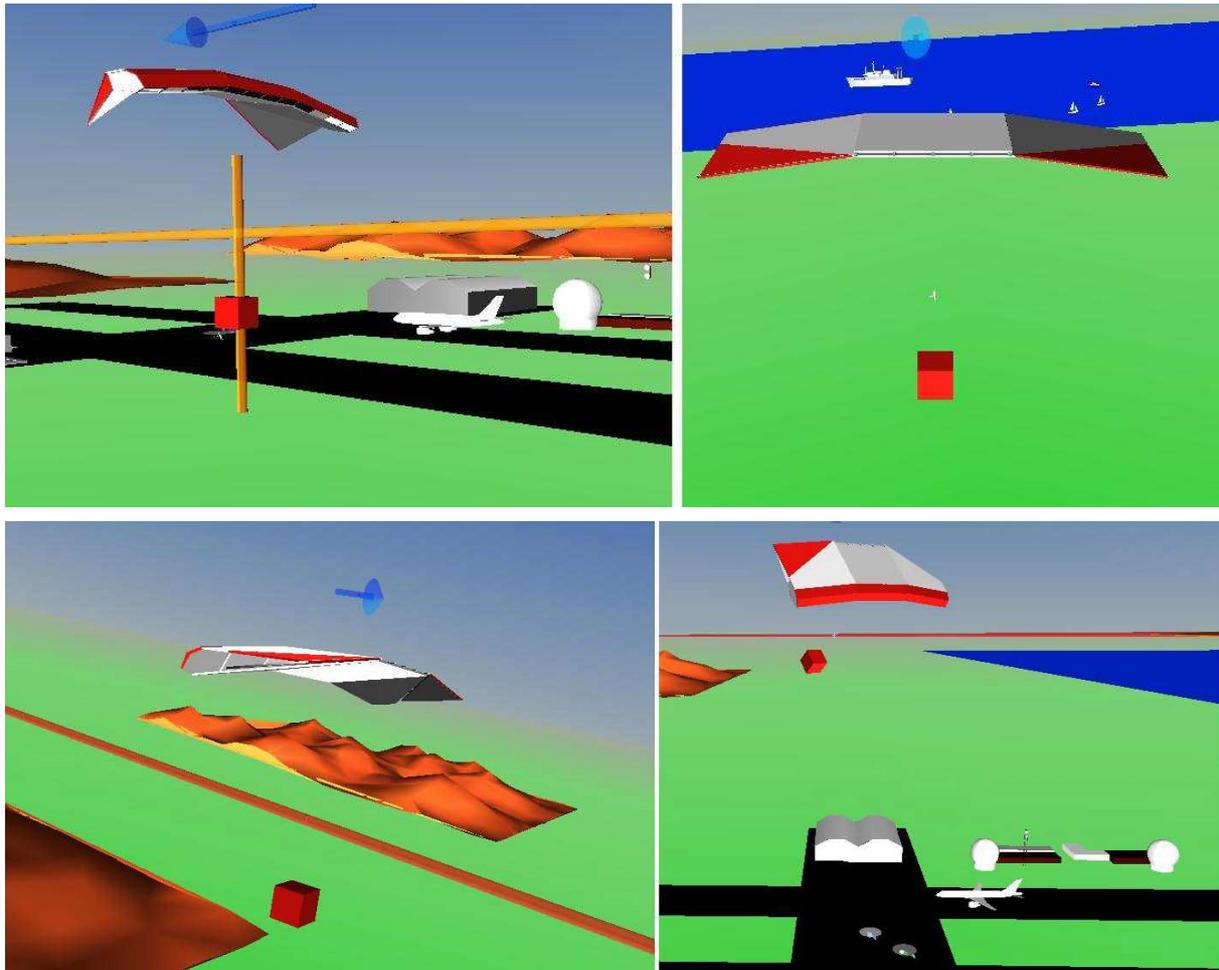


Figure 4. Several screenshots from flight of parachute model. Note the blue arrow (wind direction) and the control flaps (red triangles).

5 Conclusion

The models of flying objects, constructed by authors, serve to design and tune of automatic flight control systems. There is designed above mentioned model of parachute and also the models of an airplane and an airship. These models (some of their parts, some outputs etc.) also serve for educational purposes for students of Faculty of Mechanical engineering in such subjects as Automatic control, Simulation programming etc.

Reference

- [1] Nelson, R. C. *Flight stability and automatic control*. McGraw Hill, Boston 1998.
- [2] Stengel, R. 1989. *Flight dynamics*. Princeton university press, Princeton 2004.
- [3] Fabiani, P. A. 2007. *Autonomous flight and navigation of VTOL UAVs: form autonomy demonstrations to out-of-sight flights*. *Aerospace Science and Technology* 11/2007 s. 183-193

Ing. Hynek Havliš

Ústav přístrojové a řídicí techniky FS ČVUT, Technická 4, 166 07 Praha 6 - Dejvice

Doc. Ing. Tomáš Vyhlídal, PhD

Ústav přístrojové a řídicí techniky FS ČVUT, Technická 4, 166 07 Praha 6 - Dejvice