Skydiver

Group 7

July 25, 2013

Miriam Ruiz Ferrández
Faculty of Mathematics, University of Santiago de Compostela, Galicia, Spain
Philipp Müller
Faculty of Engineering Sciences, Tampere University of Technology, P.O. Box 692, FI-33101 Tampere, Finland
Katharina Rafetseder
Faculty of Mathematics, Johannes Kepler Universität Linz, Altenberger Str. 69, 4040 Linz, Austria
Tom Slenders
Faculty of Mathematics and Computer Science, Eindhoven University of Technology, Den Dolech 2, Postbus 513, 5600 MB Eindhoven, The Netherlands
Angelos Toytziaridis
Faculty of Engineering, University of Lund, Box 117, SE-221 00 Lund, Sweden

Instructor: K. Kulshreshtha
,University of Paderborn,
, , Germany

Abstract

Modelling the trajectory of a skydiver.

1 Introduction

In the trajectory of a skydiver there is a critical point where he should switch on his parachute. After this point consequences can be disastrous if he hasn't switched it on yet. Our aim is to predict the skydiver's height in order to the parachute switches on automatically when he is too near of the floor.
We design a model based on free fall law and wind drag force. Furthermore, we have some data about six different skydivers. The idea is to use our ODE model and the available data to predict the skydiver's height.

2 Problem formulation

During the skydiving a small device samples data every 0.25 s which is used to determine the skydiver's height and velocity. The sampled data includes a lot of noise due to the skydiver rotating and changing position all the time. Therefore the height and velocity can not be estimated accurately. So our goal is to reduce sampled noise.

3 Approach

Method 1

1. Create an ODE.
2. Estimate unknown parameters.
3. Create a extended Kalman filter.

Our idea is to

Method 2

1. Create an ODE.
2. Estimate unknown parameters.
3. Create a ARMA model.

4 Theory

4.1 Given equations

We let h be the height and v the velocity in the z-direction. Barometric formula, from which we get the velocity.

$$\frac{P}{P_0} = \left(\frac{T_0}{T_0 + L_0 h}\right)^{\frac{gM}{R^* L_0}} \tag{1}$$

Wind drag

$$D = \frac{1}{2}\varrho v^2 c_D A \tag{2}$$

Air density

$$\rho = \rho_0 \left(\frac{T_0 + L_0 h}{T_0}\right)^{\left(-\frac{gM}{R^* L_0}\right)-1} \tag{3}$$

Where the standard Temperature lapse rate $L_0 = -0.0065\,\frac{K}{m}$, standard Temperature $T_0 = 293\,K$, density $\rho_0 = 1.2041$, Molar mass of air $M = 0.0289644\,\frac{kg}{mol}$ and the Universal gas constant $R^* = 8.31432$.

4.2 Assumptions

The gravitational acceleration can be kept constant because there is only a notable change every 100 km.

4.3 ODE

We find that the z-acceleration is $\frac{d^2z}{dt^2} = \frac{D}{m} - g$, where D is the wind drag (2), $g = 9.81$ m/s^2 is the gravitational acceleration and m is the mass of the diver. After inserting we receive the following equation

$$\frac{d^2z}{dt^2} = \frac{1}{2m}\rho(z)\left(\frac{dz}{dt}\right)^2 \underbrace{c_D A}_{=:c^*(t)} - g, \tag{4}$$

with $\rho(z) = \rho_0 \left(\frac{T_0 + L_0 z}{T_0}\right)^{(-\frac{gM}{R^* L_0})-1}$. In the following we will assume that the parameter c^* is constant over the time.

4.4 Approximation of parameters

The next step is to approximate the unknown parameter $c^* := c_D A$, which depends for example on the way the diver dives. Hence, the best way is to determine this parameter by means of non-linear data fitting for ODEs. The first step is to rewrite the equation to the following system of equations

$$\frac{dz}{dt} = v(t)$$
$$\frac{dv}{dt} = \frac{1}{2m}\rho(z)v(t)^2 c^* - g.$$

Given the data points $(t_i, y_i)_{i=1}^m$ we compute the estimate of the parameter c^* such that

$$\min_{c\in\mathbb{R}} \quad \phi = \frac{1}{2}\sum_{i=1}^{m} \|\hat{y}(t_i) - y_i\|_2^2 \tag{5}$$

$$s.t. \quad \frac{d\mathbf{x}}{dt}(t) = f(t, \mathbf{x}(t), c) \quad \mathbf{x}(t_0) = \mathbf{x}_0 \tag{6}$$

$$\hat{y}(t) = g(\mathbf{x}(t), c) \tag{7}$$

$$c_l \leq c \leq c_u, \tag{8}$$

where in our case

$$\mathbf{x}(t) = \begin{bmatrix} z(t) \\ v(t) \end{bmatrix},$$

$$f(t, x(t), c) = \begin{bmatrix} v(t) \\ \frac{1}{2m}\rho(z)v(t)^2 c(t) - g \end{bmatrix},$$

$$g(\mathbf{x}(t), c) = z(t).$$

One should note that we need some initial values for the height $z(t_0)$ and the velocity $v(t_0)$.

4.5 Extended Kalman filter

We begin with writing our ODE as a system of first order ODE:s.

$$\frac{d^2h}{dt^2} = \frac{D}{m} - g \Longleftrightarrow f(h, v) = \begin{cases} \dot{h} = v & , h(t_0) = h_0 \\ \dot{v} = \frac{1}{2m}\varrho v(t)^2 c_D A - g & , v(t_0) = v_0 \end{cases}$$

The extended Kalman filter can be formulated as

$$\mathbf{x}_k = \begin{bmatrix} h_k \\ v_k \end{bmatrix} = f(\mathbf{x}_{k-1}) + \mathbf{w}_{k-1} = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_{k-1}$$
$$\mathbf{z}_k = \begin{bmatrix} h_k \\ \dot{v}_k \end{bmatrix} = c(\mathbf{x}_k) + \mathbf{v}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k$$

where $\mathbf{v}$ and $\mathbf{w}$ are the state transition and observation noises which are both assumed to be zero mean multivariate Gaussian noises with covariance $\mathbf{Q}_k$ and $\mathbf{R}_k$ respectively. In our case all of these are assumed to be zero. The process of applying the extended Kalman filter can divided into two parts predict and update.

Predict

Predicted state estimate	$\hat{\mathbf{x}}_{k\|k-1} = f(\hat{\mathbf{x}}_{k-1\|k-1})$
Predicted covariance estimate	$\mathbf{P}_{k\|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1\|k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}$

Update

Innovation or measurement residual	$\tilde{\mathbf{y}} = \mathbf{z}_k - c(\hat{\mathbf{x}}_{k\|k-1})$
Innovation (or residual) covariance	$\mathbf{S}_k = \mathbf{C}_k\mathbf{P}_{k\|k-1}\mathbf{C}_k^T + \mathbf{R}_k$
Near-optimal Kalman gain	$\mathbf{K} = \mathbf{P}_{k\|k-1}\mathbf{C}_k^T\mathbf{S}_k^{-1}$
Updated state estimate	$\hat{\mathbf{x}}_k^k = \hat{\mathbf{x}}_{k\|k-1} + \mathbf{K}_k\tilde{\mathbf{y}}_k$
Updated estimate covariance	$\mathbf{P}_{k\|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)\mathbf{P}_{k\|k-1}$

where the state transition matrix are defined to be the following Jacobians

$$\mathbf{F}_{k-1} = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k-1|k-1},\mathbf{u}_{k-1}}$$
$$\mathbf{C}_k = \left.\frac{\partial c}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k|k-1},\mathbf{u}_{k-1}}$$

and $c =$

5 ARIMA model

6 Result

7 Conclusion