



## **A Method of Visualizing Velocity and Acceleration Profiles of an Air Target**

Witold BUŻANTOWICZ

*Military University of Technology, Faculty of Mechatronics and Aerospace,  
2 gen. Witolda Urbanowicza Str., 00-908 Warsaw, Poland  
E-mail address: witold.buzantowicz@wat.edu.pl*

*Received by the editorial staff on 8 May 2018*

*The reviewed and verified version was received on 13 December 2018*

DOI 10.5604/01.3001.0012.7329

**Abstract.** This article outlines and describes the mathematical foundations for a method of generating ribbon trajectories using the Matlab environment. An example of their application is given, involving the visualization of velocity and acceleration profiles in program modules used for the planning and analysis of the flight paths of air targets.

**Keywords:** visualization, kinematics, air target, computer simulation, Matlab

### **1. INTRODUCTION**

The presentation of the flight trajectories of air targets in a ribbon form is not a new idea. In the literature, it has been used, for instance, to present the details of manoeuvring combat between two or more aircrafts [2-4]. The approach allows for a clear presentation of the nuances of the manoeuvres performed, particularly in terms of roll angles, which cannot be visualized by means of curves (Fig. 1).

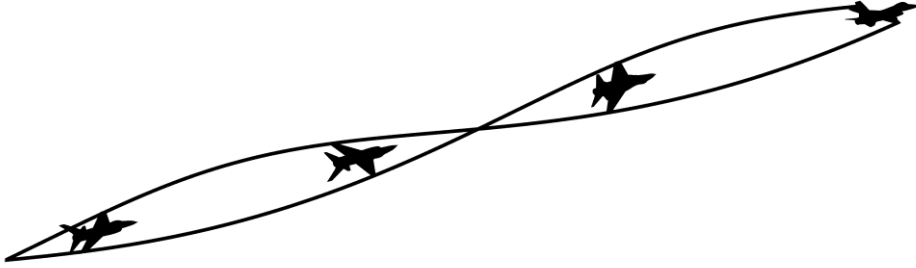


Fig. 1. An example of visualization of the spatial orientation of an object on a trajectory using a ribbon

Ribbon trajectories may be a useful way of picturing not only the orientation of an object but also its velocity and acceleration profiles at specified points in space. This expands the possibilities of graphical design, presentation and analysis of targets' flight trajectories in route editors and other tools created for use in anti-aircraft missile systems simulators.

This article proposes a method of generation and an application for ribbon trajectories in visualizing the velocity and acceleration profiles of an air target.

The article is structured as follows. Section 2 contains a model of an air target and the mathematical foundations for the description of ribbon trajectories. In section 3, an example is given of the plotting of ribbon trajectories in the Matlab environment, and key code fragments needed for this task are presented. The final section contains conclusions and observations.

## 2. MATHEMATICAL FOUNDATIONS FOR THE PLOTTING OF RIBBON TRAJECTORIES

For the purposes of plotting ribbon trajectories, a model is adopted for an air target whose motion in space is described by the vector of parameters

$$\mathbf{T} \in \mathbf{R}^{1 \times 7}: \mathbf{T} = [x \quad y \quad z \quad \theta \quad \varphi \quad \gamma \quad V] \quad (1)$$

where  $x, y, z$  are the coordinates of the target's position (m),  $\theta$  is the pitch angle of the target's velocity vector (rad),  $\varphi$  is the yaw angle of the target's velocity vector (rad),  $\gamma$  is the target's roll angle (rad), and  $V$  is the modulus of the target's velocity vector (m/s).

Changes in the target's accelerations are approximated by the first-order dynamics

$$\dot{a}_L = \frac{\kappa_L - a_L}{\tau} \quad \dot{a}_\theta = \frac{\kappa_\theta - a_\theta}{\tau} \quad \dot{a}_\varphi = \frac{\kappa_\varphi - a_\varphi}{\tau} \quad (2)$$

where  $\kappa_L$  is the target's commanded longitudinal acceleration ( $\text{m/s}^2$ ),  $\kappa_\Theta$  is the target's commanded acceleration in the pitch plane ( $\text{m/s}^2$ ),  $\kappa_\Phi$  is the target's commanded acceleration in the yaw plane ( $\text{m/s}^2$ ),  $a_L$  is the target's longitudinal acceleration ( $\text{m/s}^2$ ),  $a_\Theta$  is the target's acceleration in the pitch plane ( $\text{m/s}^2$ ),  $a_\Phi$  is the target's acceleration in the yaw plane ( $\text{m/s}^2$ ), and  $\tau$  is the time constant (s), given that the commanded accelerations are the subject to limitations in the form

$$|\kappa_L| \leq \bar{\kappa}_L \quad |\kappa_\Theta| \leq \bar{\kappa}_\Theta \quad |\kappa_\Phi| \leq \bar{\kappa}_\Phi \quad (3)$$

Changes in the target's velocity and pitch and yaw angles are defined as

$$\dot{V} = a_L \quad \dot{\theta} = \frac{a_\Theta}{V} \quad \dot{\varphi} = \frac{a_\Phi}{V} \quad (4)$$

while the target's roll angle is approximated by

$$\gamma = \arctan\left(\frac{V}{g}\dot{\varphi}\right) = \arctan\left(\frac{a_\Phi}{g}\right) \quad (5)$$

where  $g$  is the acceleration due to the Earth's gravity ( $\text{m/s}^2$ ). The following kinematic equations for the motion of the target's centre of mass are used:

$$\begin{aligned} \dot{x} &= V \cos \theta \sin \varphi \\ \dot{y} &= V \sin \theta \\ \dot{z} &= V \cos \theta \cos \varphi \end{aligned} \quad (6)$$

By means of numerical computations, using equations (1)-(6), for the commanded accelerations  $\kappa_L$ ,  $\kappa_\Theta$  and  $\kappa_\Phi$  and  $a$  defined set of initial and boundary conditions, we can determine the matrix  $\mathbf{K}$ , which is composed of three vectors  $\mathbf{x}$ ,  $\mathbf{y}$ , and  $\mathbf{z}$  of Cartesian coordinates and three vectors  $\boldsymbol{\theta}$ ,  $\boldsymbol{\varphi}$ , and  $\boldsymbol{\gamma}$  of angles:

$$\begin{aligned} \mathbf{K} \in \mathbf{R}^{j \times 6} : \mathbf{K} &= [\mathbf{x} \quad \mathbf{y} \quad \mathbf{z} \quad \boldsymbol{\theta} \quad \boldsymbol{\varphi} \quad \boldsymbol{\gamma}] \\ &= \begin{bmatrix} x_1 & y_1 & z_1 & \theta_1 & \varphi_1 & \gamma_1 \\ x_2 & y_2 & z_2 & \theta_2 & \varphi_2 & \gamma_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_j & y_j & z_j & \theta_j & \varphi_j & \gamma_j \end{bmatrix} \end{aligned} \quad (7)$$

where  $i \in \{1, 2, \dots, j\}$  is the number of the sample (instant) in discrete time,  $x_i$ ,  $y_i$ ,  $z_i$  are the Cartesian coordinates of the position of the object at the  $i$ th instant (m), and  $\theta_i$ ,  $\varphi_i$ ,  $\gamma_i$  are the pitch, yaw and roll angles of the object at the  $i$ th instant (rad). Knowing the simulation step size  $\Delta t$ , the matrix  $\mathbf{K}$  is sufficient to give a full description of the position and orientation of the target in time and space.

The curve representing the trajectory of the target (Fig. 2) is then transformed into a ribbon.

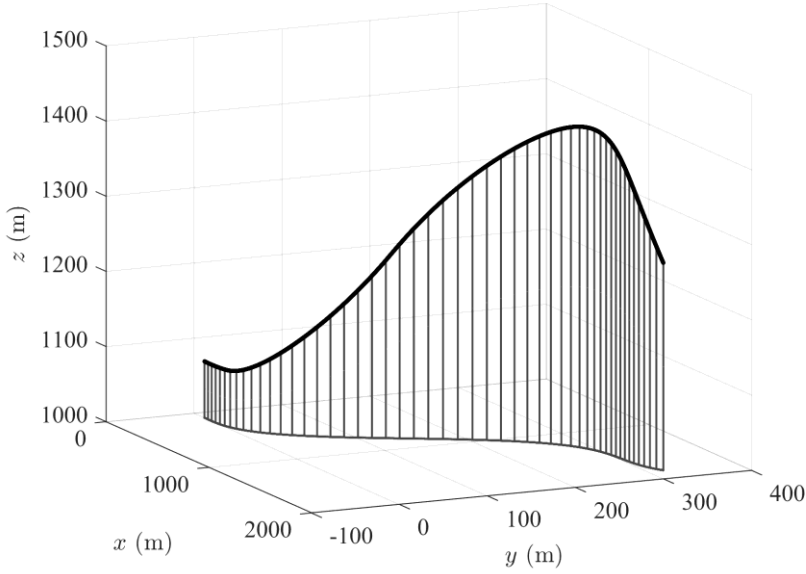


Fig. 2. Exemplary target trajectory

If the generated ribbon is to be used to visualize a velocity profile, it is necessary to compute the orientation of its individual sections, which should be in accordance with the corresponding values of the pitch angle  $\theta_i$ , yaw angle  $\varphi_i$  and roll angle  $\gamma_i$  at the considered point of the trajectory. For this purpose, a set of vectors describing the left-hand edge of the ribbon is determined:

$$\mathbf{N}_L \in \mathbf{R}^{3 \times j} : \mathbf{N}_L = \{\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_i, \dots, \mathbf{n}_j\} \quad (8)$$

in which

$$\mathbf{n}_i = \mathbf{M}_i \begin{bmatrix} 0 & \frac{1}{2}l & 0 \end{bmatrix}^T \quad (9)$$

where  $i \in \{1, 2, \dots, j\}$  is the number of the sample (instant) in discrete time,  $l$  is the width of the ribbon (m), and  $\mathbf{M}_i$  is the matrix of rotation at the  $i$ th instant, defined as

$\mathbf{M}_i \in \mathbf{R}^{3 \times 3}$  :

$$\mathbf{M}_i = \begin{bmatrix} \cos \theta_i \cos \varphi_i & -\sin \theta_i \cos \varphi_i & -\sin \varphi_i \\ \begin{pmatrix} \cos \theta_i \sin \varphi_i \sin \gamma_i \\ + \sin \theta_i \cos \gamma_i \end{pmatrix} & \begin{pmatrix} -\sin \theta_i \sin \varphi_i \sin \gamma_i \\ + \cos \theta_i \cos \gamma_i \end{pmatrix} & -\cos \varphi_i \cos \gamma_i \\ \begin{pmatrix} -\cos \theta_i \sin \varphi_i \cos \gamma_i \\ + \sin \theta_i \sin \gamma_i \end{pmatrix} & \begin{pmatrix} \sin \theta_i \sin \varphi_i \cos \gamma_i \\ + \cos \theta_i \sin \gamma_i \end{pmatrix} & \cos \varphi_i \cos \gamma_i \end{bmatrix} \quad (10)$$

Based on  $\mathbf{N}_L$ , a set of the vectors  $\mathbf{N}_R$  describing the right-hand edge of the ribbon is determined:

$$\mathbf{N}_R \in \mathbf{R}^{3 \times j} : \mathbf{N}_R = \{-\mathbf{n}_1, -\mathbf{n}_2, \dots, -\mathbf{n}_i, \dots, -\mathbf{n}_j\} \quad (11)$$

The origin of the vector  $\mathbf{n}_i$  and of the corresponding vector  $-\mathbf{n}_i$  is a point with the coordinates  $(x_i, y_i, z_i)$ , lying on the determined trajectory (Fig. 3). The curve obtained by joining the end-points of the vectors in the set  $\mathbf{N}_L$  forms, in space, the left-hand edge of the ribbon, while the curve obtained by joining the end-points of the vectors in the set  $\mathbf{N}_R$  forms the right-hand edge (Fig. 4). Based on the coordinates of the left- and right-hand edges of the ribbon, polygons are constructed, with a colour that depends on the modulus of the velocity vector  $V$  at the point  $(x_i, y_i, z_i)$  of the trajectory (Fig. 5).

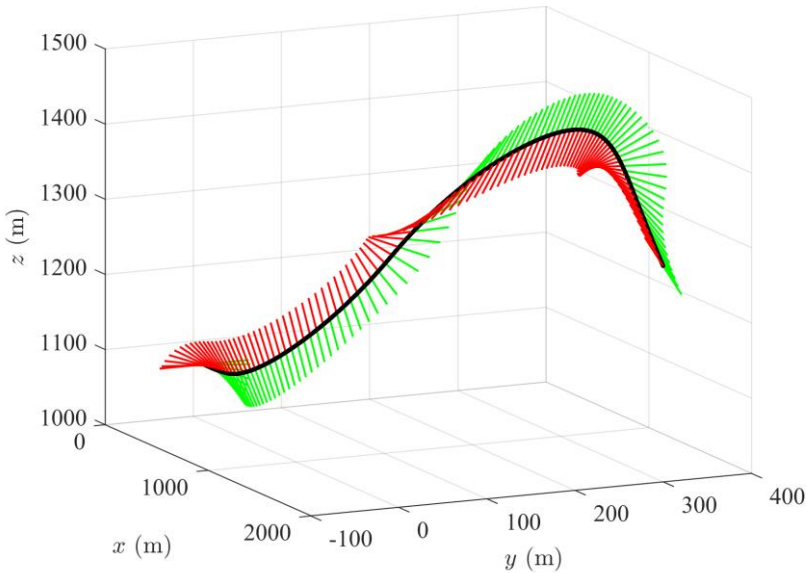


Fig. 3. Vector sets  $\mathbf{N}_L$  and  $\mathbf{N}_R$

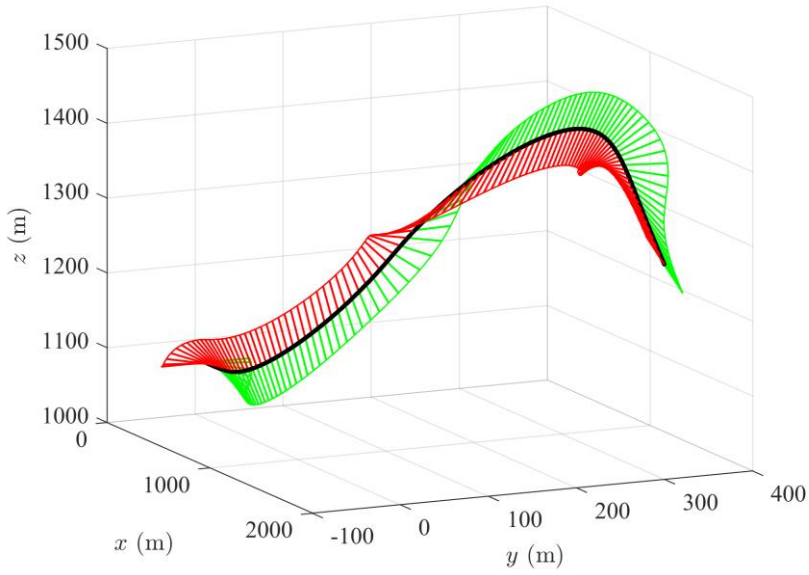


Fig. 4. Edges of the ribbon formed by joining the end-points of vectors in the sets  $\mathbf{N}_L$  and  $\mathbf{N}_R$

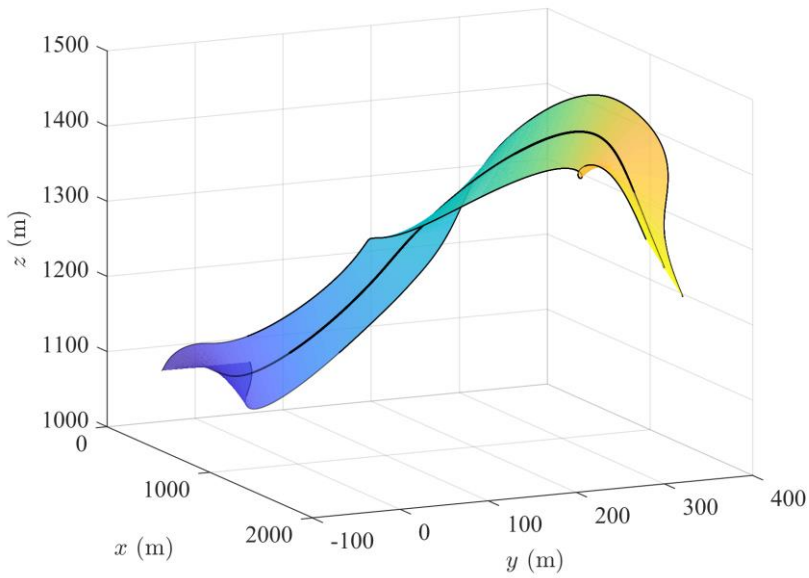


Fig. 5. A ribbon with an overlaid velocity profile

If an acceleration profile is to be visualized, the orientation of the ribbon at the point  $(x_i, y_i, z_i)$  should be in accordance with the direction and sense of the resultant acceleration vector  $\mathbf{a}_i$  at the  $i$ th instant of the simulation. For this purpose, the following set of vectors is determined:

$$\mathbf{A} \in \mathbf{R}^{3 \times j} : \mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_i, \dots, \mathbf{a}_j\} \quad (12)$$

in which

$$\mathbf{a}_i = \mathbf{M}_i [\xi a_{Li} \quad \xi a_{\phi i} \quad \xi a_{\theta i}]^T \quad (13)$$

where  $i \in \{1, 2, \dots, j\}$  is the number of the sample (instant) in discrete time,  $a_{Li}$  is the target's longitudinal acceleration at the  $i$ th instant ( $\text{m/s}^2$ ),  $a_{\theta i}$  is the target's acceleration in the pitch plane at the  $i$ th instant ( $\text{m/s}^2$ ),  $a_{\phi i}$  is the target's acceleration in the yaw plane at the  $i$ th instant ( $\text{m/s}^2$ ),  $\xi$  is the dimensionless scaling factor, and  $\mathbf{M}_i$  is the rotation matrix for the  $i$ th instant, as defined by Eq. (10).

The origin of the vector  $\mathbf{a}_i$  is the point with coordinates  $(x_i, y_i, z_i)$ , lying on the determined trajectory. The curve obtained by joining the end-points of the vectors in the set  $\mathbf{A}$  forms, in space, the outer edge of the ribbon, while the trajectory of the target forms the inner edge (Fig. 6). The width of the plotted ribbon and its colour at the point  $(x_i, y_i, z_i)$  on the trajectory depend on the modulus of the vector  $\mathbf{a}_i$  (Fig. 7).

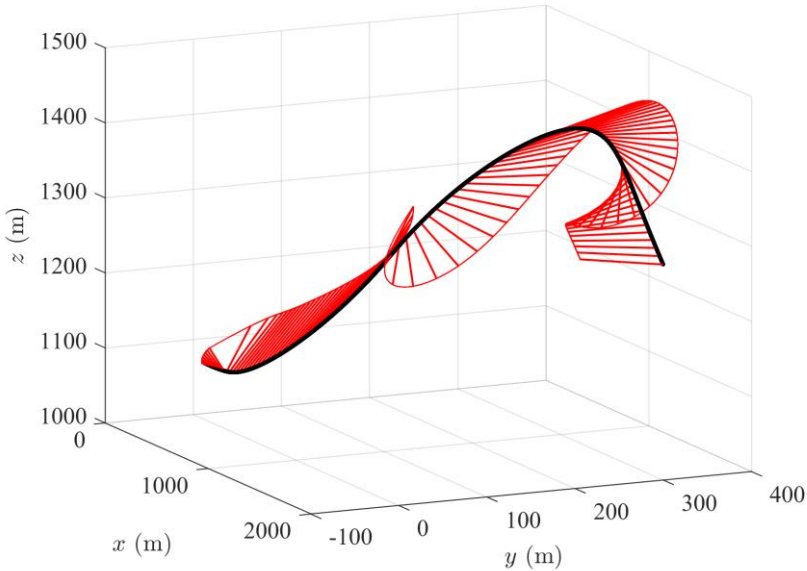


Fig. 6. Edge of a ribbon formed by joining the end-points of vectors in the set  $\mathbf{A}$

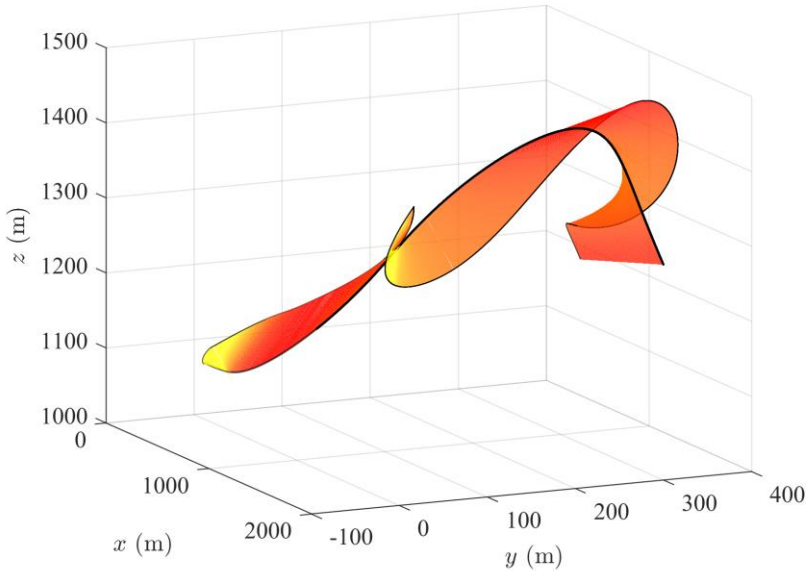


Fig. 7. A ribbon with an overlaid acceleration profile

### 3. AN EXAMPLE OF VISUALIZATION OF RIBBON TRAJECTORIES IN THE MATLAB ENVIRONMENT

Flight trajectories, together with the instantaneous position and orientation of the target in space, may be visualized in the Matlab environment using the program described in [1]. In essence, this program combines data describing the motion, geometry and other properties of objects (missiles, air targets, ground-based elements) into a single set, and uses this to generate a three-dimensional visualization of the tactical situation.

Objects are created using the *new\_object* function. An example call of this function is given in the following listing:

```
new_object('trajectory.mat',K,...
    'model','mi-24.mat','scale',10.0,...
    'edge',[.2 .2 .2],'face',[.3 .3 .3],...
    'path','on','pathcolor',[.5 .5 .5],'pathwidth',1);
```

A required input parameter for the function is the matrix  $\mathbf{K}$ , given by expression (7). The remaining parameters serve to define properties of the object or to modify the defaults, and are described in detail in the user manual. The *model* parameter can be used to select a model of an object from a resource library. The object created is saved in a file (named *trajectory.mat* in this example).



Next, the *flypath* function is called. Within the region of a chart bounded by the parameters *xlim*, *yylim* and *zlim*, this plots the trajectory and three-dimensional models with a step size defined by the *step* parameter:

```
flypath('trajectory.mat','step',100,...
        'xlim',[0 2000],'ylim',[0 600],'zlim',[1000 1500]);
```

To compute the instantaneous orientation of the object in three-dimensional space, the function uses the matrix  $\mathbf{M}_i$  defined by expression (10). Models of targets are placed on the trajectory by default. To draw an object above the ribbon (so that it is more clearly seen in the visualization) it should be translated by the vector

$$\mathbf{w} = [0 \quad 0 \quad h]^T \quad (14)$$

in the local coordinate system associated with the target

$$\mathbf{w}_i = \begin{bmatrix} w_{xi} \\ w_{yi} \\ w_{zi} \end{bmatrix} = \mathbf{M}_i \mathbf{w} \quad (15)$$

The matrix  $\mathbf{K}$  thus undergoes the transformation

$$\mathbf{K} = \begin{bmatrix} x_1 + w_{x1} & y_1 + w_{y1} & z_1 + w_{z1} & \theta_1 & \varphi_1 & \gamma_1 \\ x_2 + w_{x2} & y_2 + w_{y2} & z_2 + w_{z2} & \theta_2 & \varphi_2 & \gamma_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_j + w_{xj} & y_j + w_{yj} & z_j + w_{zj} & \theta_j & \varphi_j & \gamma_j \end{bmatrix} \quad (16)$$

and this is the form that should be provided as the input parameter to the *new\_object* function.

The sets of coordinates *x*, *y*, and *z* of the left- and right-hand edges of the ribbon, stored in the Matlab environment in the form of the vectors *xr*, *yr* and *zr*:

```
% x : [1xj] x-coordinate vector of trajectory points
% y : [1xj] y-coordinate vector of trajectory points
% z : [1xj] z-coordinate vector of trajectory points
% n : [3xj] ribbon edge vector set, where
%     (1,:) are x-coordinates of vectors' end-points,
%     (2,:) are y-coordinates of vectors' end-points,
%     (3,:) are z-coordinates of vectors' end-points
xr = [x+n(1,:); x-n(1,:)];
yr = [y+n(2,:); y-n(2,:)];
zr = [z+n(3,:); z-n(3,:)];
```

are used as input data to the *patch* function, which generates polygons based on arrays of coordinates of vertices *vx*, *vy* and *vz*:

```

c = [v;fliplr(v)]; % define a color for each vertex
                    % using target velocity vector values

vx = [x,fliplr(xr(1,:))]; % x-coordinates of vertex list
vy = [y,fliplr(yr(1,:))]; % y-coordinates of vertex list
vz = [z,fliplr(zr(1,:))]; % z-coordinates of vertex list
patch(vx,vy,vz,c);        % draw left half-ribbon
vx = [x,fliplr(xr(2,:))]; % x-coordinates of vertex list
vy = [y,fliplr(yr(2,:))]; % y-coordinates of vertex list
vz = [z,fliplr(zr(2,:))]; % z-coordinates of vertex list
patch(vx,vy,vz,c);        % draw right half-ribbon

```

As the above listing shows, the resulting ribbon is plotted using two half-ribbons: the left half-ribbon, contained between the edge formed by the end-points of vectors in the set  $\mathbf{N}_L$  and the trajectory, and the right half-ribbon, contained between the trajectory and the edge formed by the end-points of vectors in the set  $\mathbf{N}_R$ . In the case of the visualization of an acceleration profile, an analogous procedure is followed, except that only one ribbon is plotted, contained between the outer edge formed by the end-points of vectors in the set  $\mathbf{A}$  and the inner edge described by the trajectory. Figures 8 and 9 show exemplary results of visualization of the velocity and acceleration profiles of an air target using the described procedure.

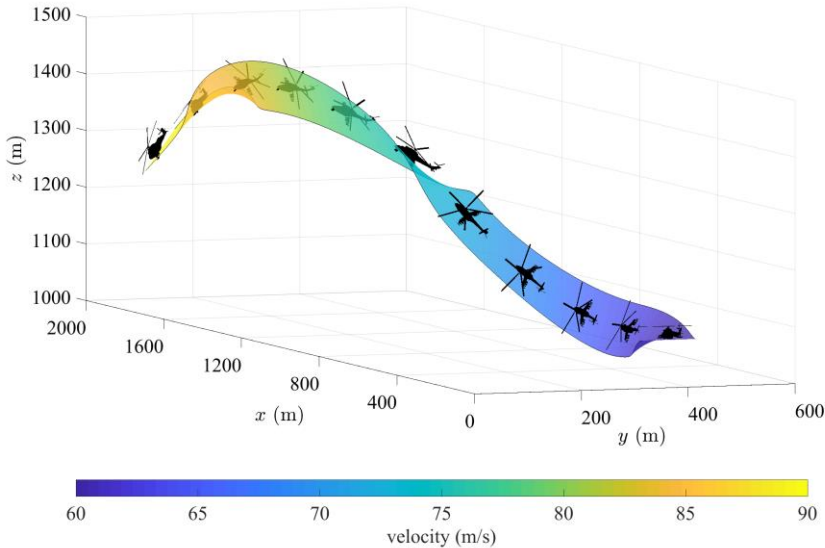


Fig. 8. Ribbon trajectory with a velocity profile

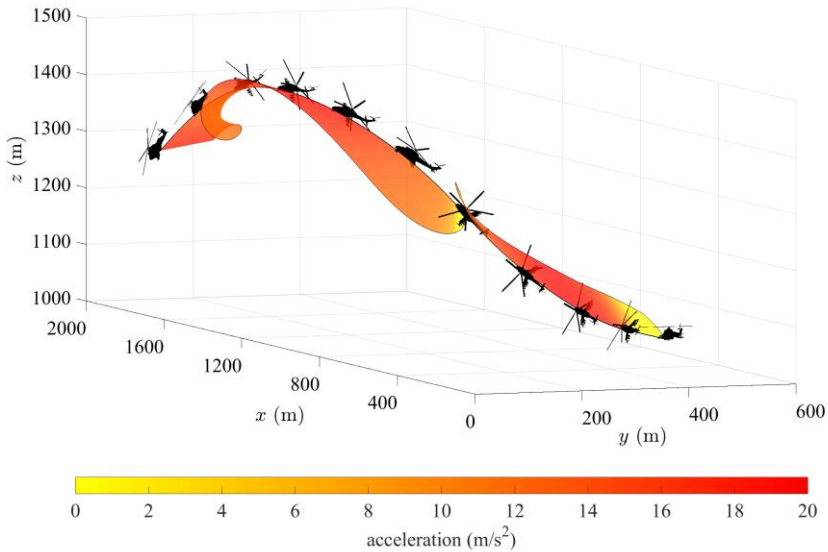


Fig. 9. Ribbon trajectory with an acceleration profile

## 4. CONCLUSIONS

This paper has outlined and given the mathematical foundations for a method of generating ribbon trajectories using the Matlab environment. Presentation of the trajectories of air targets, using a coloured ribbon with variable width may be a useful technique for the visualization of additional information about an object (velocities, accelerations, dynamic pressure, aerodynamic forces and moments, angles of attack, etc.) at every point on its flight path. This is an important functionality for program modules serving to edit flight paths and to plan the flight trajectories of targets, which are used, for example, in anti-aircraft missile systems simulators.

## REFERENCES

- [1] Bużantowicz Witold. 2016. “Matlab Script for 3D Visualization of Missile and Air Target Trajectories”. *International Journal of Computer and Information Technology* 5(5) : 419-422.
- [2] Shaw L. Robert. 1985. *Fighter Tactics: Combat and Maneuvering*. Annapolis: Naval Institute Press.
- [3] Spick Mike. 1987. *An Illustrated Guide to Modern Fighter Combat*. Upper Saddle River: Prentice Hall.
- [4] Spick Mike. 1983. *Fighter Pilot Tactics: The Techniques of Daylight Air Combat*. New York: Stein and Day.

## O pewnym sposobie obrazowania profilów prędkości i przyspieszenia celu powietrznego

Witold BUŻANTOWICZ

*Wojskowa Akademia Techniczna, Wydział Mechatroniki i Lotnictwa,  
ul. gen. Witolda Urbanowicza 2, 00-908 Warszawa*

**Streszczenie.** W artykule przedstawiono matematyczne podstawy opisu i zarys sposobu generowania trajektorii wstęgowych za pomocą środowiska Matlab. Jako przykład ich zastosowania wskazano wizualizacje profilów prędkości i przyspieszenia w modułach programowych przeznaczonych do planowania i analizowania tras lotu celów powietrznych.

**Słowa kluczowe:** wizualizacja, kinematyka, cel powietrzny, symulacja komputerowa, Matlab