How to cite this article:

Authors: Jacek Warchulski, Marcin Warchulski

Title of article: „Process automatization of creating shape of anti-aircraft missile nose in AutoCAD software”

Mechanik, No. 7 (2019)DOI: <https://doi.org/10.17814/mechanik.2019.7.54>

Process automatization of creating shape of anti-aircraft missile nose in AutoCAD software

JACEK WARCHULSKI**MARCIN WARCHULSKI***

* Mjr dr inż. Jacek Warchulski (jacek.warchulski@wat.edu.pl), <https://orcid.org/0000-0003-4451-5711> – Wydział Mechatroniki i Lotnictwa, Wojskowa Akademia Techniczna, Warszawa, Polska
 Mjr dr inż. Marcin Warchulski (marcin.warchulski@wat.edu.pl), <https://orcid.org/0000-0003-1310-3001> – Wydział Mechatroniki i Lotnictwa, Wojskowa Akademia Techniczna, Warszawa, Polska

In this paper possibilities of using of CAx systems to automation process of graphical tasks are presented. Possibility of using Delphi language and ActiveX technology to creating the shape of an anti-aircraft missile in CAx system are analyzed.

KEYWORDS: programming, Delphi, ActiveX, AutoCAD, 3D objects

Introduction

The proper use of a large amount of energy from a rocket engine, transformed in a relatively short time, requires ensuring the most favorable movement conditions. Rocket armament designers therefore use optimization methods as widely as possible. Two issues play a major role in the search for optimal flight conditions. First of all, optimal control programs should be selected, including propulsion programs and rocket direction control programs, and secondly - rocket shapes with the lowest aerodynamic resistance [6].

Basic criteria in rocket flight dynamics are the minimum time to reach the target and the minimum energy needed to travel the route to the target.

When considering the motion of the projectile in the atmosphere, the minimum-time criterion does not take into account energy losses, which are the greater the faster the rocket's flight speed. Increasing the speed should result in a shorter flight time. This does not mean, however, that in the case of missiles controlled by a minimal-time program, energy losses play no role. However, the only way to reduce them is to reduce aerodynamic resistance. Therefore, searching for the shape of the body with the least aerodynamic resistance is one of the most important problems.

The most favorable rocket hull contour can be sought for various ranges of flight speed. Different speed ranges correspond to different air reflection hypotheses. The search for the optimal nose contour can be reduced to minimizing pressure resistance without taking into account friction. However, friction resistance can only be considered when determining the most favorable nose extension. This approach is justified, because frictional forces as tangent to the stroke slightly affect its shape, but are important in determining the slenderness of the nose.

The function describing the nature of the anti-aircraft rocket nose curve can be determined using a variation calculus. The task is to determine the contour resistance of the expanding part of the hull defined as the rotary body (fig. 1).

Fig. 1. Elemental resistance force distribution (d - nose diameter, ds - elementary path, x - independent variable)

Function describing the nose shape with minimum aerodynamic resistance is defined as follows:

$$f(x) = \frac{d}{2} \left(\frac{x}{l_n} \right)^{\left(\frac{n}{n+1} \right)}$$

where: x - independent variable, d - nose diameter, l_n - nose length.

For the function defined in this way and for the shape of the cylindrical part of the hull (fig. 2), the total resistance coefficient of the anti-aircraft rocket is the sum of the pressure resistance coefficient C_{xc} and friction resistance C_{xt} [6]:

$$C_x = C_{xc} + C_{xt} = \frac{16}{2^{(n+1)}} \cdot \frac{1}{\lambda_n^{(n-1)}} \left(\frac{n}{n+1} \right)^n \cdot k + 4C_t \left(\frac{n+1}{2n+1} \cdot \lambda_n + \lambda_c \right)$$

where: λ_n - extension of the nose, λ_c - extension of the cylindrical part.

Fig. 2. Optimization of the shape of an anti-aircraft rocket for supersonic flow ($n = 3$ and $k = \frac{4}{\sqrt{M^2 - 1}}$, C_t - coefficient of friction of a flat plate depending on the Reynolds number)

Access to AutoCAD objects

Free modeling systems do not contain functions that automatically generate the coordinates of nodal points of polylines or regions, which are in turn used to create solid objects. For these reasons, it was decided to develop an application that would generate the geometry of the spatial model of the nose of the anti-aircraft rocket using AutoCAD and a free programming environment.

Programming of graphics tasks is possible using the automation interface and programming languages built into the CAx systems [1-3]. The requirements set in the process of generating the anti-aircraft rocket model are fulfilled by the client-server interface, with the role of the automation client being played in Delphi using the Embarcadero RAD Studio programming environment, while the role of the automation server - AutoCAD 2018.

OLE automation includes the ability to programmatically control the objects of another program and the protocol, by which a program can access an object in another program. Connecting OLE server to the AutoCAD is possible due to the CreateOleObject function, which when invoked, creates an object sharing AutoCAD objects. However, to gain access to the currently open drawing, the GetActiveOleObject function is used, which provides objects of the active external program. Connection to the OLE server for programs created in the Delphi language is provided by the PolaczAutoCAD module (fig. 3). The PolaczZACAD procedure in this module allows for access to the space of the model of the CAx system running.

TABLE. Selected methods of the OLE automation interface

Syntax	Application
RetVal := object.AddLine(StartPoint, EndPoint) <i>*RetVal - Return a Value</i>	Draws a segment between two points
RetVal := object.AddLightweightPolyline (VerticesList) <i>* VerticesList - matrix of vertices (array of Doubles - multiple of the number)</i>	Draws a 2D polyline based on the given list of vertices
RetVal := object.AddRegion (ObjectList) <i>* ObjectList - array of objects forming the region (arcs, circles, lines, ellipses, polylines or splines)</i>	Creates a region from a set of components (objects must form a closed area)
RetVal := object.AddRevolvedSolid (Profile, AxisPoint, AxisDir, Angle) <i>* Profile - region type object * AxisPoint - coordinate starting point of the axis of rotation * AxisDir - 3D vector specifying the direction of the axis of rotation * TaperAngle - angle in radians</i>	Draws a revolving solid based on the given profile and rotation axis

```

unit PolaczAutoCAD; {nazwa modułu}
interface           {deklaracje widoczne dla innych programów i modułów}
uses               {lista modułów zawierających importowane elementy}
  ComObj, Variants, Dialogs; {moduł obsługi obiektów OLE i Dialogs}
var               {deklaracje zmiennych}
  AcadApp, AcadDoc, AcadDwg :OLEVariant;
  {pod zmienną AcadApp będzie przypisany obiekt Autocad.Application}
  {pod zmienną AcadDoc będzie przypisany obiekt Autocad.Application.Document}
  {a pod zmienną AcadDwg obiekt Autocad.Application.Document.ModelSpace}
function PolaczZACAD :Boolean; {deklaracja funkcji}
implementation     {deklaracje niewidoczne dla innych programów i modułów}
function PolaczZACAD :Boolean;
begin
  Result:=False;    {przypisanie wartości funkcji PolaczZACAD}
  try               {początek zewnętrznego bloku chronionego}
    try            {początek wewnętrznego bloku chronionego}
      {jeśli AutoCAD jest uruchomiony}
      AcadApp := GetActiveOLEObject('AutoCAD.Application');
    except         {jeśli AutoCAD jest uruchomiony to należy stworzyć obiekt}
      AcadApp := CreateOLEObject('AutoCAD.Application');
      AcadApp.Visible:=True;
    end;          {koniec wewnętrznego bloku chronionego}
    AcadDoc:=AcadApp.ActiveDocument;
    if AcadApp.Documents.Count > 0 then
    begin         {rysunek przypisujemy go do zmiennej AcadDwg}
      AcadDwg:=AcadDoc.ModelSpace;
      Result:=True; {jeśli uzyskano dostęp do obiektu przyjmujemy wartość True}
    end
    else        {jeśli nie jest otworzony to wyświetlony zostanie komunikat}
      ShowMessage('Najpierw proszę założyć lub otworzyć jakiś dokument...');
    except     {w razie niepowodzenia w dostępie od obiektu AutoCAD}
      ShowMessage('Brak aktywnych rysunków lub aktywne polecenie.');
```

Fig. 3. Module code PolaczAutoCAD [4]

Example of automating the introduction of spatial objects

The RAD Studio 10.2 Tokyo Starter programming environment and Delphi programming language were used to develop the computer program Nosek.exe [4], enabling the introduction in the AutoCAD model area of a uniform solid nose of the anti-aircraft rocket.

```

var Nosek : OLEVariant;
{ ... }
//dodanie 2 segmentów (0,0) do (ln,0) i (koniec funkcji) do (ln,0)
Polilinia_Zarys[ind_poli]:=AcadDwg.AddLine(p2D(x1,y1), p2D(x1,0));
Polilinia_Zarys[ind_poli+1]:=AcadDwg.AddLine(p2D(x1,0), p2D(0,0));
//utworzenie regionu i bryły obrotowej
Region1:=AcadDwg.AddRegion(Polilinia_Zarys);
Nosek:=AcadDwg.AddRevolvedSolid(Region1[0], p2D(0,0), p2D(x1,0), 2*Pi);
//Usuwanie obwiedni
for i:=0 to ind_poli+1 do Polilinia_Zarys[i].Delete;
Region1[0].Delete;
//Zoom
AcadDoc.Application.ZoomExtents;
{ ... }
```

Fig. 4. Fragment of the Nosek.pas program code - creating a block [4]

Fig. 5. Operation of the Nosek.exe program in AutoCAD [4]

A rocket nose block can be created using standard spatial drawing commands, based on profiles defining a given block (both polyline and region objects can be used).

To automate this process, one must define selected methods of the AutoCAD 2018 OLE interface (see table) for entering flat and spatial objects [5].

Function describing the shape of the nose was used to create a closed profile of the region type, on the basis of which a solid object was created using the AddRevolvedSolid command. A fragment of the code of the main module Nosek.pas, which allows for defining a solid by twisting the region and removing the twisting envelope, is shown in fig. 4.

The operation of the Nosek.exe program is shown in fig. 5.

Summary

Users can automate repetitive design tasks with computer-aided design programs. For this purpose, special, newly developed programs can be used, which are not standard attached to CAD programs [7-11].

Automatic definition of AutoCAD drawing components can have a wide practical use. The example of using the Delphi language presented in the paper confirms that the use of programming languages allows for the automation of modeling of flat and spatial objects in computer-aided systems, which indirectly also leads to a more productive use of given CAX systems.

REFERENCES

- [1] Dudek M. „AutoLISP. Praktyczny kurs”. Helion, 1997.
- [2] Czyżycki W., Lisowski E. „Automatyzacja zadań grafiki za pomocą Delphi”. Helion, 2002.
- [3] Warchulski J., Warchulski M. „Przykłady automatyzacji zadań grafiki w programie AutoCAD”. XIV Międzynarodowa Szkoła Komputerowego Wspomagania Projektowania, Wytwarzania i Eksploatacji, Jurata. (10–14 maja 2010): 515–522.
- [4] Warchulski J., Warchulski M. „Program Nosek.exe”. Warszawa: Wojskowa Akademia Techniczna, 2019.
- [5] <https://knowledge.autodesk.com/search-result/caas/CloudHelp/cloudhelp/2016/ENU/AutoCAD-ActiveX/files/GUID-5D302758-ED3F-4062-A254-FB57BAB01C44-htm.html>.
- [6] Dubiel S. „Uogólniona postać rozwiązania obrysu noska osiowo-symetrycznego o najmniejszym oporze aerodynamicznym”. *Mechanika Teoretyczna i Stosowana*. 28, 3–4 (1990): 513–524.
- [7] Warchulski J., Warchulski M. „Wykorzystanie programu AutoCAD do generowania punktów węzłowych polilinii”. *Mechanik*. 7 (2014): 713–720.
- [8] Warchulski J., Warchulski M. „Wykorzystanie programu AutoCAD do generowania kształtu rakiety przeciwlotniczej”. *Mechanik*. 7 (2015): 931–942. DOI: <http://dx.doi.org/10.17814/mechanik.2015.7.314>.
- [9] Warchulski J., Warchulski M. „Automatyzacja użycia bloków z atrybutami w programie AutoCAD”. *Mechanik*. 7 (2016): 862–863. DOI: <http://dx.doi.org/10.17814/mechanik.2016.7.209>.
- [10] Warchulski J., Warchulski M. „Automatyzacja definiowania kształtu profili lotniczych w programie AutoCAD”. *Mechanik*. 7 (2017): 609–611. DOI: <https://doi.org/10.17814/mechanik.2017.7.88>.
- [11] Warchulski J., Warchulski M. „Automatyzacja definiowania obiektów przestrzennych w programie AutoCAD”. *Mechanik*. 7 (2018): 564–566. DOI: <https://doi.org/10.17814/mechanik.2018.7.85>.