Dr inż. Jacek WARCHULSKI
Dr inż. Marcin WARCHULSKI
Wojskowa Akademia Techniczna

DOI: 10.17814/mechanik.2015.7.314

WYKORZYSTANIE PROGRAMU AUTOCAD DO GENEROWANIA KSZTAŁTU RAKIETY PRZECIWLOTNICZEJ

Streszczenie: W referacie zaprezentowano możliwości wykorzystania programu AutoCAD do generowania kształtu rakiety przeciwlotniczej. Przedstawiono możliwość wykorzystania języka programowania Delphi w procesie automatyzacji zadań grafiki.

USE OF AUTOCAD SOFTWARE TO GENERATE THE SHAPE OF THE ANTI-AIRCRAFT MISSILE

Abstract: The paper presents the possibility of using AutoCAD to generate the shape of the anti-aircraft missile. Possibility of using Delphi programming language in the process of automating tasks graphics was presented.

Słowa kluczowe: AutoCAD, rakieta przeciwlotnicza, Delphi
Keywords: AutoCAD, anti-aircraft missile, Delphi

1. WPROWADZENIE

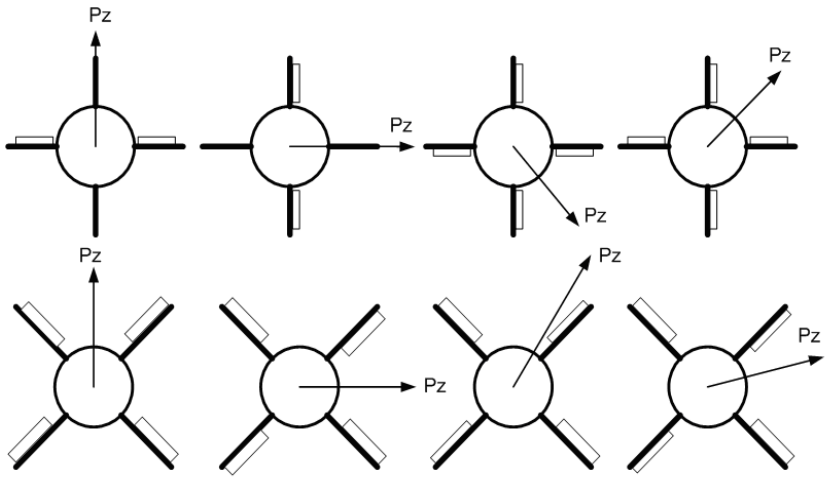
Różnorodność zadań stawianych rakietom przeciwlotniczym implikuje istotne różnice w konstrukcji ich płatowca. Obejmuje to kształt zewnętrzny kadłuba, a także skrzydeł i usterzenia oraz ich rozmieszczenie i wzajemne położenie względem siebie. Sposób rozmieszczenia i umiejscowienie powierzchni nośnych na kadłubie klasyfikuje rakiety na charakterystyczne układy o określonych właściwościach w widoku z przodu (z tyłu), z boku i z góry. Warianty rozmieszczenia powierzchni nośnych rakiety dają w efekcie dwa rodzaje układów: osiowo-symetryczny oraz płasko-symetryczny [5].

Układ osiowo-symetryczny stosuje się w rakietach, od których wymaga się zdolności sterowania przestrzennego za pomocą dwukanałowego sygnału sterowania. Odnosi się to do zmieniania się kierunku lotu rakiety w zależności od panującej sytuacji powietrznej w czasie realizacji zadania bojowego. Umiejscowienie powierzchni nośnych w układzie osiowo-symetrycznym (przechylenia powierzchni nośnych) może być względnie dowolne. Zasadniczo wyróżnia się dwa usytuowania, tj. układ krzyżowy, kiedy jedna para powierzchni nośnych leży w płaszczyźnie poziomej, a druga w pionowej. W przypadku przechylenia powierzchni nośnych względem płaszczyzny poziomej i pionowej mamy do czynienia z układem „X”.

Spośród rakiet sterowanych aerodynamicznie wyróżnia się dwa rodzaje powierzchni nośnych: skrzydła i usterzenie. Płaszczyzny ich umiejscowienia występują w dwóch konfiguracjach: pokrywającej się ze sobą lub mogą być przechylone względem siebie. Wynikiem tego będą różne widoki z przodu (z tyłu). Wymieniając w pierwszej kolejności umiejscowienie

powierzchni nośnych skrzydeł, a w drugiej usytuowanie powierzchni nośnych usterzenia, można określić następujące układy osiowo-symetryczne w widoku z przodu: układ (+ +), układ (+ X), układ (X X), układ (X +), układ (+ X) lub (X +), układ o powierzchniach nośnych rozmieszczonych co 120°, układ (+) z silnikami startowymi.

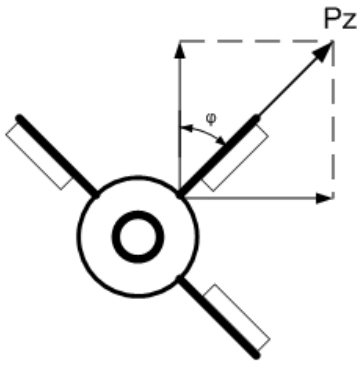
Najistotniejszą zaletą układów osiowo-symetrycznych jest możliwość sterowania przy pomocy dwóch sygnałów wpływających na kąt wznoszenia i kąt kierunku lotu, zwanych inaczej kątami elewacji i azymutu. Sposób takiego sterowania przedstawiono na rysunku 1.



Rys. 1. Generowanie siły sterującej odpowiednim wychyleniem sterów

Układ płasko-symetryczny stosowany jest czasami przy rakietach dalekiego zasięgu z małym zużyciem paliwa. Główne powierzchnie nośne spełniają rolę płaskich skrzydeł i wytwarzają siłę nośną równoważącą masę rakiety (rysunek 2). Główna część toru lotu tego typu rakiet jest torem prostoliniowym, zwykle poziomym, pochylonym o niewielki kąt w dół. Taki lot nazywany jest lotem ślizgowym. Wykorzystywanie płaskiego układu skrzydeł pozwala na znaczne zwiększenie zasięgu rakiety, stosując ten sam zespół napędowy. Przykładowo rakieta balistyczna o zasięgu 200-300 km, wyposażona w skrzydła umożliwiające lot ślizgowy, może zwiększyć zasięg nawet trzykrotnie.

Jest także możliwość sterowania rakietą podczas lotu ślizgowego sterami aerodynamicznego. Jednakże komplikuje się poważnie system sterowania. Aby w pełni można było manewrować taką rakietą, potrzebny jest trzykanałowy system przekazywania sygnałów odpowiedzialny za sterowanie wysokością, kierunkiem i lotkami.



Rys. 2. Generowanie siły nośnej w układzie płasko-symetrycznym

Przykładem rakiety z zastosowaniem układu płasko-symetrycznego jest rakieta BGM-109 Tomahawk (rysunek 3).

Rys. 3. Rakieta BGM-109 Tomahawk [6]

Rakiety osiowo-symetryczne różnią się w widoku z przodu od samolotu, czyli od układu płasko-symetrycznego. W widoku z boku ta różnica nie występuje, dlatego nazwa układu rakiet w tym widoku została adaptowana w odniesieniu do samolotów. Można wyróżnić cztery typowe układy: układ klasyczny, układ typu „kaczka", układ bezogonowy (bezogonowiec) oraz układ z wychylanymi skrzydłami (rysunek 4).

a) Układ klasyczny – rakieta zestawu S-75M WOŁCHOW [7]

b) Układ typu „kaczka" – rakiety zestawu S-125 NEWA SC na wyrzutni [8]

c) Układ bezogonowy – rakieta zestawu S-200 WEGA [9]

d) Układ z wychylanymi skrzydłami – rakieta zestawu 2K12 KUB [10]

Rys. 4. Układy pocisków rakietowych w widoku z boku [7, 8, 9, 10]

Poszczególne układy są wynikiem doskonalenia aerodynamicznego obiektów latających, na początku dotyczącego lotnictwa, ale z biegiem czasu przejętego również przez technikę rakietową. W zależności od przeznaczenia rakiety stosuje się inny układ aerodynamiczny. Wiąże się to z wykonywanym zadaniem i systemem sterowania. Nieokreślone jest, który układ będzie najlepszym rozwiązaniem. W różnych przypadkach przeważają zalety raz jednego, a raz drugiego.

2. MODELOWANIE GEOMETRII RAKIETY PRZECIWLOTNICZEJ

W procesie opracowywania modelu matematyczno-fizycznego rakiety przeciwlotniczej zachodzi potrzeba określenia jej struktury, a w szczególności charakterystyk geometrycznych oraz masowo-bezwładnościowych. W procesie określania charakterystyk geometrycznych rakiety przeciwlotniczej można skorzystać z dokumentacji technicznej, opisów technicznych czy też obiektu rzeczywistego. Większej ilości pracy wymaga natomiast określenie charakterystyk masowo-bezwładnościowych, ponieważ rakieta przeciwlotnicza to obiekt o nieregularnych kształtach i zróżnicowanym rozkładzie mas.

Proces modelowania ruchu rakiety przeciwlotniczej wymaga wyznaczenia położenia środka masy rakiety oraz wartości momentów bezwładności względem środka masy. Wartości te występują w skalarnych równaniach dynamicznych ruchu pocisku rakietowego wokół środka masy:

$$I_x \cdot \frac{dp}{dt} - (I_y - I_z) \cdot q \cdot r = L \tag{1}$$

$$I_y \cdot \frac{dq}{dt} - (I_z - I_x) \cdot p \cdot r = M \tag{2}$$

$$I_z \cdot \frac{dr}{dt} - (I_x - I_y) \cdot p \cdot q = N \tag{3}$$

gdzie:

- p, q, r – składowe prędkości kątowej Ω rakiety w układzie związanym z rakietą 0xyz wzdłuż osi x, y, z;
- I_x, I_y, I_z – momenty bezwładności rakiety względem osi: x, y, z układu współrzędnych związanego z rakietą 0xyz;
- L, M, N – odpowiednio: moment przechylający, pochylający i odchylający względem środka masy pocisku rakietowego.

Z powyższych względów zdecydowano się na wykonanie aplikacji, która umożliwiłaby generowanie geometrii modelu rakiety przeciwlotniczej z wykorzystaniem programu AutoCAD na potrzeby wyznaczania charakterystyk masowo-bezwładnościowych.

Realizacja programowania zadań grafiki możliwa jest w oparciu o interfejs automatyzacji *OLE* oraz języki programowania wbudowane w systemy CAx [1, 2, 3]. Interfejsem spełniającym wymagania postawione w procesie generowania modelu rakiety przeciwlotniczej jest interfejs typu klient−serwer, gdzie rolę klienta automatyzacji pełni program napisany w języku *Delphi* z wykorzystaniem narzędzia programowego *Embarcadero RAD Studio XE3*, natomiast rolę serwera automatyzacji pełni program *AutoCAD 2015*.

Nawiązanie połączenia z serwerem *OLE* programu AutoCAD możliwe jest dzięki wykorzystaniu funkcji *CreateOleObject*, która po wywołaniu tworzy obiekt udostępniający obiekty programu AutoCAD. Natomiast do uzyskania dostępu do aktualnie otwartego rysunku wykorzystuje się funkcję *GetOleObject*, która udostępnia obiekty aktywnego programu zewnętrznego. Uzyskanie połączenia z serwerem *OLE* dla programów w języku Delphi przedstawiono w module *PolaczAutoCAD* (rysunek 5). Procedura *PolaczZACAD* znajdująca się w tym module pozwala uzyskać dostęp do uruchomionego obiektu aplikacji.

W artykule przedstawiono także niezbędne funkcje translacji współrzędnych wykorzystywane przez interfejs automatyzacji *OLE*.

```
unit PolaczAutoCAD;   //nazwa modułu
                      //część opisowa modułu
interface             //deklaracje widoczne dla innych programów i modułów
uses                  //lista modułów zawierających importowane elementy
  ComObj, Variants;   //moduł obsługi obiektów OLE
var                   //deklaracje zmiennych
  AcadApp, AcadDoc, AcadDwg:OleVariant;
                      //AcadApp - obiekt Autocad.Application,
                      //AcadDoc - obiekt Autocad.Application.Document
                      //AcadDwg - obiekt Autocad.Application.Document.ModelSpace
function PolaczZACAD :Boolean;
implementation        //deklaracje niewidoczne dla innych programów i modułów
uses  Dialogs;
                     //implementacja zdefiniowanych procedur i funkcji
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
      //jeśli jest otworzony jakiś rysunek to przypisujemy go do zmiennej AcadDwg
      AcadDwg:=AcadDoc.ModelSpace;
      //jeśli uzyskano dostęp do objektu drawing przyjmuje się wartość True
      Result:=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.');
     //MessageDlg('Brak aktywnych rysunków' +
     //           ' lub aktywne polecenie.',mtError,[mbOK], 0);
  end; //koniec zewnętrznego bloku chronionego
end;
initialization
finalization
  AcadApp:=Unassigned;  AcadDoc:=Unassigned;  AcadDwg:=Unassigned;
end.
```

Rys. 5. Kod modułu połączenia z programem AutoCAD [4]

Program AutoCAD określa położenie punktu w przestrzeni przy pomocy trzech współrzędnych (x, y, z). Interfejs automatyzacji *OLE* programu AutoCAD wymaga podania współrzędnych w postaci właściwej trójelementowej tablicy, której elementami są liczby typu *Double*. Jednak taki typ tablic nie może być stosowany, gdy wykorzystujemy mechanizm *OLE*. Do przesyłania współrzędnych użyto zmiennych typu *OLEVariant*. Można je przekształcić na typ tablicowy przy pomocy funkcji *VarArrayCreate*. Przekształcenie to przedstawiono na rysunku 6.

```
function p2D(x,y:Double) :OleVariant;  //zwraca wariantową tablicę liczb Double
  Begin                                //w postaci wektora [x,y,0]
      //przeksztalcenie zmiennej typu OLEVariant w tablicę liczb Double
    Result:=varArrayCreate([0,2], varDouble);
      //przypisywanie wartości elementom tablicy
    Result[0]:=x;  Result[1]:=y;  Result[2]:=0;
  end;
```

Rys. 6. Kod funkcji przekształcania współrzędnych [4]

Niektóre polecenia wymagają podania tablicy współrzędnych punktów o nieokreślonej liczbie elementów, w szczególności przy generowaniu elementów takich jak *polilinia* czy *region*. Taka tablica musi być zgodna z typem *OLEVariant*. W celu skrócenia zapisu i sprawdzenia poprawności wprowadzanych danych, jak również przekazywania pojedynczych punktów, opracowano odpowiednią funkcję zwalniającą użytkownika z definiowania tablicy wariantowej. Jej zapis przedstawiono na rysunku 7.

```
function vTab3D(WspXYZ:array of double) :OleVariant;
      //zwraca wariantową tablicę punktów w przestrzeni
      //ilość liczb musi być wielokrotnością trzech
var  i:Integer;
  begin
    if (High(WspXYZ)+1) mod 3 = 0 then
      begin
        Result:=VarArrayCreate([0,High(WspXYZ)],varDouble);
        for i:=0 to High(WspXYZ) do Result[i]:=WspXYZ[i];
      end
    else
      ShowMessage('Nie zawiera wszystkich współrzędnych punktów!');
  end;
```

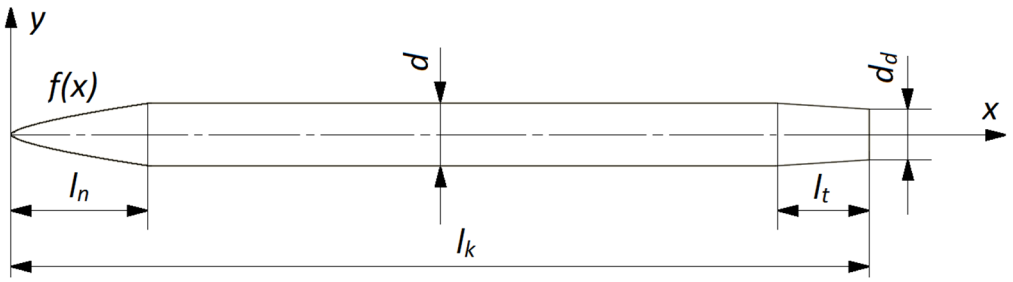
Rys. 7. Kod funkcji przekształcania współrzędnych o nieokreślonej liczbie elementów [4]

Tworząc program do automatyzacji generowania kształtu rakiety przeciwlotniczej w systemie CAx, konieczne było przyjęcie określonych założeń ograniczających, gdyż zróżnicowanie kształtów rakiet przeciwlotniczych jest bardzo duże. W związku z powyższym sformułowane zostały następujące założenia projektowe:

- generowany model będzie przedstawiał rakietę jednostopniową,
- część nosowa kadłuba rakiety będzie definiowana jako kształt dający minimalny opór aerodynamiczny w zależności od zakresu prędkości,
- generowany model będzie zawierał dwie pary powierzchni nośnych – cztery powierzchnie sterowe oraz cztery skrzydła (stateczniki),

- przyjęto osiowo-symetryczny układ aerodynamiczny – skrzydła i stery ustawione będą względem siebie w konfiguracji (+ +);
- rozmieszczenie powierzchni nośnych: pierwsze powierzchnie nośne mogą być usytuowane na części nosowej bądź cylindrycznej kadłuba rakiety, natomiast drugie tylko na części cylindrycznej;
- biorąc pod uwagę gabaryty płatowca rakiety przeciwlotniczej dokładność wprowadzania modelu została określona na 0,5 [mm].

Podstawowe parametry geometryczne kadłuba rakiety przeciwlotniczej przedstawiono na rysunku 8.



Rys. 8. Wymiary geometryczne kadłuba rakiety przeciwlotniczej

Gdzie:
l_k – długość całkowita kadłuba;
l_n – długość części nosowej;
l_t – długość części tylnej;
d – średnica kadłuba;
d_d – średnica części tylnej.

Kształt obrysu noska opisuje funkcja, która została wyznaczona za pomocą rachunku wariacyjnego. Wykorzystując hipotezy odbicia niutonowskiego, lustrzanego oraz gazu nieściśliwego, utworzona została funkcja opisująca opór czołowy, która posłużyła do zbudowania funkcjonału. Dla tego typu funkcji argumentami są wektory, natomiast wartościami wielkości skalarne. Istotą rachunku wariacyjnego jest znalezienie takich ekstremali (funkcji), dla których funkcjonał osiąga ekstremum, w tym wypadku dla wartości minimalnej funkcjonał przyjmuje postać równania Eulera–Lagrange'a. Rozwiązaniem tego równania jest funkcja, dla której opór czołowy jest najmniejszy. W postaci bezwymiarowej przyjmuje ona postać:

$$\eta = \xi^{\frac{n}{n+1}}, \tag{4}$$

gdzie:
n = 2 – dla opływu naddźwiękowego;
n = 3 – dla opływu hipersonicznego;
n = 4 – dla opływu poddźwiękowego.

Podstawiając:

$$\xi = \frac{x}{\lambda_n d}, \qquad \eta = \frac{2y}{d}, \tag{5}$$

gdzie wydłużenie noska λ_n jest równe:

$$\lambda_n = \frac{l_n}{d}. \tag{6}$$

Funkcja f(x) opisująca kształt obrysu noska przyjmie następującą postać:

$$f(x) = \frac{d}{2}\left(\frac{x}{l_n}\right)^{\frac{n}{n+1}}. \tag{7}$$

Wyżej wymienioną zależnością $f(x)$ opisane są punkty węzłowe noska rakiety, gdzie argumenty tej funkcji przyjmują wartości co 0,5 [mm] od 0 do l_n. Następnym etapem określania zarysu kadłuba było zdefiniowanie połączeń pomiędzy kolejnymi jego charakterystycznymi punktami: $(l_n, \frac{d}{2})$, $(l_k - l_t, \frac{d}{2})$, $(l_k, \frac{d_d}{2})$, $(l_k, 0)$, $(0,0)$ (rysunek 9).

```
Polilinia_Zarys[ind_poli]:=AcadDwg.AddLine(p2D(x1,y1), p2D(x2,y2));
Polilinia_Zarys[ind_poli+1]:=AcadDwg.AddLine(p2D(x2,y2), p2D(x3,y3));
Polilinia_Zarys[ind_poli+2]:=AcadDwg.AddLine(p2D(x3,y3), p2D(x4,y4));
Polilinia_Zarys[ind_poli+3]:=AcadDwg.AddLine(p2D(x4,y4), p2D(0,0));
Region1:=AcadDwg.AddRegion(Polilinia_Zarys);
```

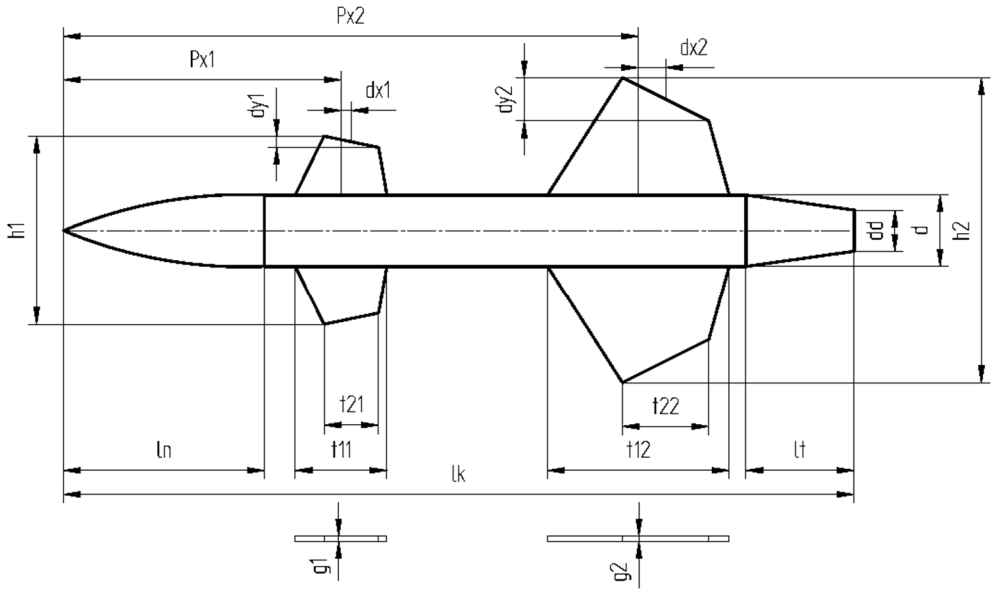
Rys. 9. Kod programu definiujący region zarysu kadłuba [4]

Powstały w ten sposób obiekt w środowisku programu AutoCAD jest obiektem typu region. Na rysunku 10 przedstawiono instrukcję tworzącą bryłę 3D kadłuba rakiety:

```
Kadlub:=AcadDwg.AddRevolvedSolid(Region1[0], p2D(0,0), p2D(x4,0), 2*Pi);
```

Rys. 10. Instrukcja tworząca kadłub rakiety przeciwlotniczej [4]

Wymiary geometryczne powierzchni nośnych oraz ich rozmieszczenie przedstawiono na rysunku 11.



Rys. 11. Wymiary geometryczne opierzenia rakiety

gdzie:
P_{x1} – odległość środka dolnej podstawy pierwszej powierzchni nośnej od noska rakiety;
h_1 – rozpiętość pierwszych powierzchni nośnych;
t_{11} – pozioma długość dolnej podstawy pierwszej powierzchni nośnej;
t_{21} – pozioma długość górnej podstawy pierwszej powierzchni nośnej;
dx_1 – przesunięcie poziome środka dolnej podstawy pierwszej powierzchni nośnej względem środka górnej podstawy;
dy_1 – różnica wysokości pierwszej powierzchni nośnej;
g_1 – grubość pierwszej powierzchni nośnej;
P_{x2} – odległość środka dolnej podstawy drugiej powierzchni nośnej od noska rakiety;
h_2 – rozpiętość drugich powierzchni nośnych;
t_{12} – pozioma długość dolnej podstawy drugiej powierzchni nośnej;
t_{22} – pozioma długość górnej podstawy drugiej powierzchni nośnej;
dx_2 – przesunięcie poziome środka dolnej podstawy drugiej powierzchni nośnej względem środka górnej podstawy;
dy_2 – różnica wysokości drugiej powierzchni nośnej;
g_2 – grubość drugiej powierzchni nośnej.

Ponadto odległość drugich powierzchni nośnych leżących naprzeciwko siebie zawsze będzie równa średnicy kadłuba. Natomiast odległość pierwszych powierzchni nośnych będzie uzależniona od usytuowania na kadłubie, a mianowicie jeżeli będą rozmieszczone na części nosowej, to ich wzajemna odległość opisana będzie funkcją zarysu noska. Jeżeli pierwsze powierzchnie nośne umiejscowione będą na części cylindrycznej, to odległość pomiędzy nimi będzie równa średnicy kadłuba tak jak w przypadku drugich powierzchni nośnych.

Pojedynczą powierzchnię nośną można zdefiniować jako prostopadłościan o wierzchołkach podstawy w punktach: $(P_{x2} - \frac{t_{12}}{2}, \frac{d}{2}, \frac{g_2}{2})$, $(P_{x2} + dx_2 - \frac{t_{12}}{2}, \frac{d}{2} + \frac{h_2}{2}, \frac{g_2}{2})$, $(P_{x1} + \frac{t_{12}}{2}, \frac{d}{2} - dy_2, \frac{g_2}{2})$, $(P_{x2} + \frac{t_{12}}{2}, \frac{d}{2}, \frac{g_2}{2})$, wysokości g_2 oraz bokach odchylonych od pionu o kąt 0°. Na rysunku 12 przedstawiono kod programu dodający bryłę powstałą przez projekcję z przekroju określonego ww. punktami tworzącymi obiekt typu *region*.

```
//drugie powierzchnie nośne
boki2:=varArrayCreate([0,3],varDispatch);
przekroj2:=varArrayCreate([0,3], varDispatch);
boki2[0]:=AcadDwg.AddLine(p3D(Px2-t12/2,d/2,g2/2), p3D(Px2+dx2-t22/2, d/2+(h2-
d)/2,g2/2));
boki2[1]:=AcadDwg.AddLine(p3D(Px2+dx2-t22/2,d/2+(h2-d)/2,g2/2),
p3D(Px2+dx2+t22/2,d/2+(h2-d)/2-dy2,g2/2));
boki2[2]:=AcadDwg.AddLine(p3D(Px2+dx2+t22/2,d/2+(h2-d)/2-dy2,g2/2),
p3D(Px2+t12/2,d/2,g2/2));
boki2[3]:=AcadDwg.AddLine(p3D(Px2+t12/2,d/2,g2/2), p3D(Px2-t12/2,d/2,g2/2));
przekroj2:=AcadDwg.AddRegion(boki2);
skrzydlo12:=AcadDwg.AddExtrudedSolid(przekroj2[0],-g2,0);
```

Rys. 12. Kod programu definiujący powierzchnię nośną na części cylindrycznej kadłuba [4]

Powstałą w ten sposób bryłę można kilkukrotnie skopiować, jednocześnie rozmieszczając odpowiednio wokół kadłuba − tworząc całą sekwencję czterech powierzchni nośnych. Wykonać to można przy pomocy funkcji *copy* tworzącej kopię wskazanego obiektu oraz

metody *Rotate3D*, która powoduje obrót wywołanego obiektu względem osi przechodzącej przez dwa punkty o podany kąt (rysunek 13).

```
skrzydlo22:=skrzydlo12.;
skrzydlo22.Rotate3D(p3D(Px1-t21/2,0,0), p3D(Px1+t21/2,0,0),Pi/2);
skrzydlo32:=skrzydlo12.Copy;
skrzydlo32.Rotate3D(p3D(Px1-t21/2,0,0), p3D(Px1+t21/2,0,0),Pi);
skrzydlo42:=skrzydlo12.Copy;
skrzydlo42.Rotate3D(p3D(Px1-t21/2,0,0), p3D(Px1+t21/2,0,0),3*Pi/2);
```

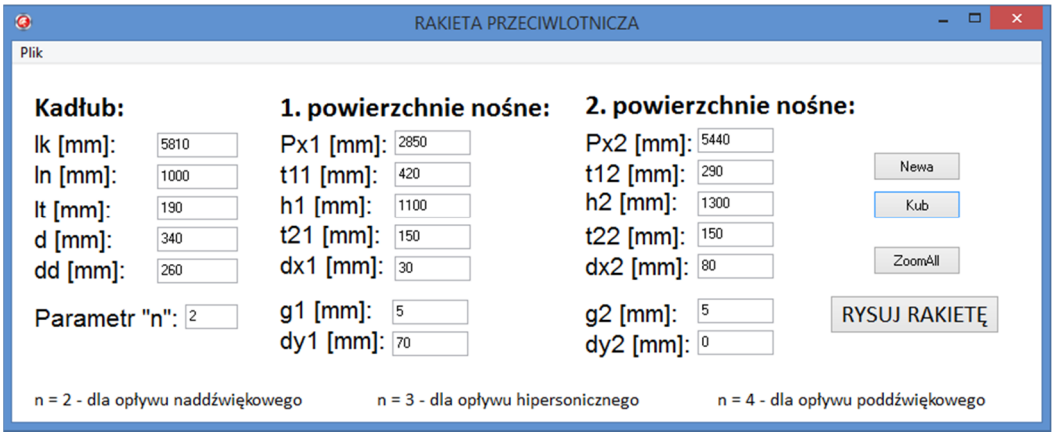
Rys. 13. Kod programu wykorzystujący funkcję *copy* oraz *Rotate3D* [4]

Postępując analogicznie, można zdefiniować pierwsze powierzchnie nośne dla przypadku usytuowania ich na części cylindrycznej kadłuba. Nieco inaczej wygląda określenie ich przy rozmieszczeniu w części nosowej kadłuba (rysunek 14).

```
boki1:=varArrayCreate([0,3], varDispatch);   //pierwsze powierzchnie nośne w części nosowej
przekroj1:=varArrayCreate([0,3], varDispatch);
boki1[0]:=AcadDwg.AddLine(p3D(Px1-t11/2, d/(2*exp(n/(n+1)*ln(lln)))*exp(n/(n+1)*ln(Px1-
t11/2)),g1/2),p3D(Px1+dx1-t21/2, h1/2, g1/2));
boki1[1]:=AcadDwg.AddLine(p3D(Px1+dx1-t21/2, h1/2,g1/2),p3D(Px1+dx1+t21/2,h1/2-dy1,g1/2));
boki1[2]:=AcadDwg.AddLine(p3D(Px1+dx1+t21/2,h1/2dy1,g1/2),p3D(Px1+t11/2,d/(2*exp(n/(n+1)*ln
(lln)))*exp(n/(n+1)*ln(Px1+t11/2)), g1/2));
boki1[3]:=AcadDwg.AddLine(p3D(Px1+t11/2,d/(2*exp(n/(n+1)*ln(lln)))*exp(n/(n+1)*ln(Px1+t11/2
)),g1/2), p3D(Px1-t11/2,d/(2*exp(n/(n+1)*ln(lln)))*exp(n/(n+1)*ln(Px1-t11/2)), g1/2));
przekroj1:=AcadDwg.AddRegion(boki1);
skrzydlo11:=AcadDwg.AddExtrudedSolid(przekroj1[0],-g1,0);
skrzydlo21:=skrzydlo11.Copy;
skrzydlo21.Rotate3D(p3D(Px1-t21/2,0,0),p3D(Px1+t21/2,0,0),Pi/2);
skrzydlo31:=skrzydlo11.Copy;
skrzydlo31.Rotate3D(p3D(Px1-t21/2,0,0),p3D(Px1+t21/2,0,0),Pi);
skrzydlo41:=skrzydlo11.Copy;
skrzydlo41.Rotate3D(p3D(Px1-t21/2,0,0),p3D(Px1+t21/2,0,0),3*Pi/2);
```

Rys. 14. Kod programu definiujący powierzchnie nośne na części nosowej kadłuba [4]

Rysunek 15 przedstawia okno główne opracowanego programu do definiowania geometrii rakiety przeciwlotniczej.



Rys. 15. Okno główne programu *Rakieta przeciwlotnicza* [4]

3. PODSUMOWANIE

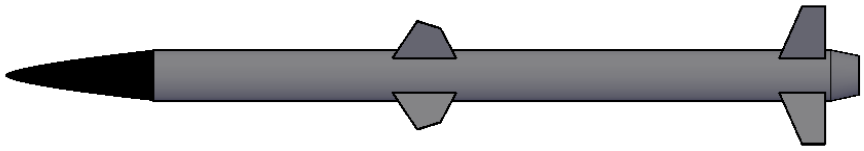
W celu sprawdzenia poprawności działania programu oraz skonfrontowania wygenerowanych modeli rakiet z obiektami rzeczywistymi, do programu wprowadzono przykładowe dane geometryczne rakiet przeciwlotniczych występujących na wyposażeniu Sił Zbrojnych RP.

Jako pierwszy został wygenerowany model rakiety przeciwlotniczego zestawu rakietowego 2K12 KUB. Dane wejściowe do programu dla tej rakiety przedstawiono w tabeli 1.

Tabela 1. Parametry wejściowe dla rakiety zestawu 2K12 KUB

Kadłub		1) powierzchnie nośne		2) powierzchnie nośne	
lk [mm]	5810	Px1 [mm]	2850	Px2 [mm]	5440
ln [mm]	1000	t11 [mm]	420	t12 [mm]	290
lt [mm]	190	h1 [mm]	1100	h2 [mm]	1300
d [mm]	340	t21 [mm]	150	t22 [mm]	150
dd [mm]	260	dx1 [mm]	30	dx2 [mm]	80
Parametr „n”	2	g1 [mm]	5	g2 [mm]	5
		dy1 [mm]	70	dy2 [mm]	0

Na rysunku 16 zobrazowano model rakiety zestawu 2K12 KUB wygenerowany przez program AutoCAD.



Rys. 16. Rakieta zestawu 2K12 KUB wygenerowana w programie AutoCAD

Jako drugi został wygenerowany model rakiety przeciwlotniczego zestawu rakietowego S-125 NEWA SC. Dane wejściowe do programu dla tej rakiety przedstawiono w tabeli 2.

Tabela 2. Parametry wejściowe dla rakiety zestawu S-125 NEWA SC

Kadłub		1. powierzchnie nośne		2. powierzchnie nośne	
lk [mm]	4130	Px1 [mm]	950	Px2 [mm]	3320
ln [mm]	2670	t11 [mm]	250	t12 [mm]	1040
lt [mm]	220	h1 [mm]	470	h2 [mm]	1110
d [mm]	380	t21 [mm]	60	t22 [mm]	280
dd [mm]	300	dx1 [mm]	90	dx2 [mm]	380
Parametr „n”	2	g1 [mm]	5	g2 [mm]	5
		dy1 [mm]	0	dy2 [mm]	0

Na rysunku 17 przedstawiono model drugiego stopnia rakiety zestawu S-125 NEWA SC wygenerowany przez program AutoCAD.

Rys. 17. Drugi stopień rakiety S-125 NEWA SC wygenerowany w programie AutoCAD

Zaprezentowany w artykule przykład użycia języka *Delphi* pokazuje, że wykorzystanie języków programowania oraz systemów *CAx* pozwala na zamodelowanie dowolnej geometrii w systemach komputerowego wspomagania.

Na podstawie automatycznie zdefiniowanej geometrii rakiet przeciwlotniczych można oszacować parametry masowo-bezwładnościowe pocisków rakietowych poprzez wykorzystanie funkcji *PARAMFIZ* dostępnej jako standardowa funkcja programu AutoCAD.

LITERATURA

[1] Czyżycki W., Lisowski E.: *Automatyzacja zadań grafiki za pomocą Delphi*, Helion, 2002.
[2] Dudek M.: *AutoLISP. Praktyczny kurs*, Helion, 1997.
[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, Materiały konferencyjne, s. 515-522.
[4] Warchulski J., Warchulski M.: *Program „Rakieta_przeciwlotnicza.exe”*, Wojskowa Akademia Techniczna, Warszawa, 2015.
[5] A. Dębecki, S. Dubiel: *Konstrukcja rakiet. Część III – Podstawy projektowania, charakterystyki aerodynamiczne i optymalne programy lotu rakiet*, Warszawa, 1988.
[6] *http://www.ausairpower.net/USN/000-TLAM-600-029.jpg*
[7] *http://upload.wikimedia.org/wikipedia/commons/f/f4/S-75M_Wolchow.jpg*
[8] *http://pvo.guns.ru/images/expo/mspo2003/s125-2.jpg*
[9] *http://www.samolotypolskie.pl/uploads/Products/product_2442/prev_s-200-wega _1.jpg*
[10] *http://www.palba.cz/forumfoto/albums/userpics/13295/KUB_4.jpg*