

Automatyzacja definiowania kształtu profili lotniczych w programie AutoCAD

Process automatization of creating aerofoil shape in AutoCAD software

JACEK WARCHULSKI
MARCIN WARCHULSKI*

DOI: <https://doi.org/10.17814/mechanik.2017.7.88>

Przedstawiono zastosowania systemów CAx w automatyzacji zadań graficznych. Przeanalizowano możliwość wykorzystania języka AutoLISP i środowiska Visual LISP do tworzenia kształtu profili lotniczych w systemie CAx.

SŁOWA KLUCZOWE: programowanie, AutoLISP, Visual LISP, AutoCAD, przekrój lotniczy

Possibilities of using of CAx systems to automation process of graphical tasks are presented. Possibility of using AutoLISP language with VisualLISP environment to create aerofoil shape in CAx system are analysed.

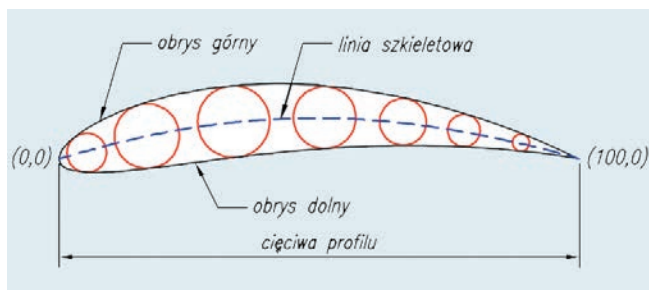
KEYWORDS: programming, AutoLISP, Visual LISP, AutoCAD, aerofoil

Program AutoCAD ma wielu użytkowników i zajmuje znaczące miejsce w graficznym zapisie konstrukcji. Można w nim programować czynności związane z obsługą danych, automatyzacją zadań graficznych oraz wykonywaniem rysunku i dzielenie informacje o nim z innymi aplikacjami.

Napisanie programu do realizacji szczegółowego zadania zwykle wymaga większego nakładu pracy, jednak później można wielokrotnie wykorzystać utworzoną aplikację [3]. Tworzenie kodów automatyzujących zadania graficzne dla programu AutoCAD w środowisku Windows jest możliwe za pomocą wbudowanych narzędzi programistycznych (np. środowiska programowania Visual LISP opartego na języku AutoLISP) [1, 2, 5, 6].

Obrys płata nośnego

Tradycyjnie profile lotnicze rysuje się w płaszczyźnie XY. Na rys. 1 wyjaśniono podstawowe pojęcia używane w opisie geometrii profili lotniczych [7].



Rys. 1. Geometria profilu lotniczego

Z lewej strony, w punkcie o współrzędnych (0,0), znajduje się zawsze tzw. nos profilu, określający kształt krawędzi natarcia płata. Nos jest zazwyczaj zaokrąglony (czasami podawany jest promień tego zaokrąglenia). Profil jest zakończony w punkcie o współrzędnych (100,0). Linia profilu jest często dzielona na dwie części: obrys górny i obrys dolny. Obrysy łączą się na krańcach profilu. Prosty odcinek, łączący nos z krawędzią spływu, nazywany jest cięciwą. Drugą charakterystyczną linią profilu lotniczego jest linia szkieletowa. Punkty tej linii leżą w równej odległości od górnej i dolnej krawędzi profilu. Można ją sobie wyobrazić jako linię przechodzącą przez środki okręgów wpisanych w obrys profilu. Dla profilu lotniczego definiuje się także takie parametry, jak maksymalna grubość czy punkt maksymalnego ugięcia linii szkieletowej.

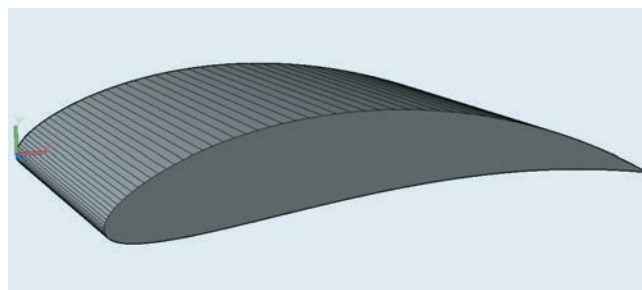
Generator kształtu profilu lotniczego

Aby zbudować model profilu lotniczego, niezbędne jest zdefiniowanie poszczególnych punktów profilu płata. W procesie tym można wykorzystać internetowe generatory profili lotniczych, np. profili z „czterocyfrowej rodziny” NACA [8, 9]. Generator profili lotniczych umożliwia zapisanie współrzędnych punktów węzłowych profilu w postaci plików tekstowych.

Numer czterocyfrowego profilu NACA zawiera opis jego kluczowych parametrów geometrycznych [7]:

- pierwsza cyfra: największe ugięcie szkieletowej (czyli wartość maksymalnego ugięcia wyrażona procentowo),
- druga cyfra: odległość punktu o największym ugięciu szkieletowej od nosa profilu podzielona przez 10 (np. wartość „2” oznacza, że maksymalne ugięcie szkieletowej znajduje się 20% od krawędzi natarcia),
- trzecia i czwarta cyfra: grubość profilu (wyrażona procentowo).

Przykładowo: symbol profilu „NACA 8516” oznacza profil o grubości 16% i maksymalnym ugięciu szkieletowej 8%, w odległości równej 50% długości cięciwy.



Rys. 2. Model profilu „NACA 8516” zdefiniowany w programie AutoCAD

* Mjr dr inż. Jacek Warchulski (jacek.warchulski@wat.edu.pl), mjr dr inż. Marcin Warchulski (marcin.warchulski@wat.edu.pl) – Wydział Mechatroniki i Lotnictwa Wojskowej Akademii Technicznej

Przykład automatyzacji wprowadzania kształtu profilu lotniczego

Język programowania AutoLISP i środowisko Visual LISP programu AutoCAD zostały wykorzystane do opracowania nowego polecenia *Profil*, umożliwiającego wprowadzenie w obszarze modelu programu AutoCAD obrysu profilu NACA, którego współrzędne znajdują się w pliku z rozszerzeniem *.txt. Kod programu *Profil.lsp* przedstawiony jest na rys. 3.

```
(defun C:Profil()
  (vl-load-com)
  (setq _acad_ (vlax-get-acad-object)
        _model_ (vla-get-ModelSpace (vla-get-ActiveDocument _acad_)))
  ; -----
  (setq plik_nazwa (getfiled "Wskaz nazwę plików z danymi:" "D:\\" "txt" 8))
  (setq plik_nazwa (findfile plik_nazwa))
  (setq plik (open plik_nazwa "r"))
  (setq linia (read-line plik))
  ; -----
  (setq dane_z_pliku (list))
  (while linia
    (setq dane (read (strcat "(" linia "))))

    (setq dane_z_pliku (cons dane dane_z_pliku))
    (setq linia (read-line plik))
  );while
  (setq dane_z_pliku (reverse dane_z_pliku))
  (close plik)
  (setq PETLA (length dane_z_pliku))
  ; -----
  (princ "\n")
  (princ "Wybierz typ wprowadzanych obiektów: \n")
  (princ "1 - PUNKTY \n")
  (princ "2 - SEGMENTY LINIOWE \n")
  (princ "3 - POLILINIA 2D \n")
  (princ "4 - POLILINIA 2D DOPASOWANA DO SPLAJNU \n")

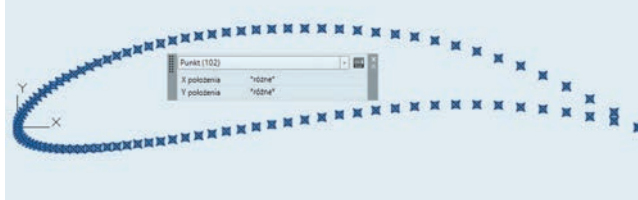
  (princ "5 - SPLAJN \n")
  (princ "WYBIERZ 1,2,3,4 lub 5 i naciśnij ENTER")
  (princ)
);defun
```

Rys. 3. Kod główny programu *Profil.lsp* języka AutoLISP [4]

Program *Profil.lsp* działa w następujący sposób:

- tworzone jest „nowe” polecenie programu AutoCAD – *Profil*, które nie ma żadnych parametrów wejściowych,
- funkcja *vl-load-com* pozwala na wywołanie funkcji pozwalających na programowanie obiektowe; do zmiennej globalnej *_acad_* przypisywany jest dokument programu AutoCAD, natomiast do zmiennej *_model_* przypisywana jest przestrzeń modelu aktywnego rysunku,
- użycie funkcji *getfiled* oraz *findfile* pozwala na wskazanie pliku z danymi oraz zdefiniowanie ścieżki dostępu do tego pliku,
- użycie funkcji *open*, *read-line* oraz *strcat* pozwala na otwarcie pliku w trybie tylko do odczytu, odczytanie poszczególnych linii danych, połączenie ich w listę i zapis do tablicy danych,

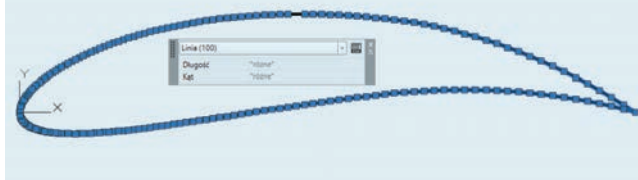
```
(defun c:1() ;PUNKTY
  (setvar "pdmode" 67)
  (setq i 0)
  (repeat (- PETLA 1)
    (setq element (nth i dane_z_pliku))
    (setq P_x (car element))
    (setq P_y (cadr element))
    (setq P_z 0)
    (vla-AddPoint _model_ (vlax-3D-Point P_x P_y
P_z))
    (setq i (+ i 1))
  );repeat
  (command "_zoom" "_e")
  (command "_regen")
);defun
```



Rys. 4. Kod podprogramu wprowadzającego punkty węzłowe profilu lotniczego

```
(defun c:2() ;SEGMENTY LINIOWE
  (setq i 0)
  (repeat (- PETLA 2)
    (setq element1 (nth i dane_z_pliku))
    (setq P1_x (car element1))

    (setq P1_y (cadr element1))
    (setq P1_z 0)
    (setq element2 (nth (+ i 1) dane_z_pliku))
    (setq P2_x (car element2))
    (setq P2_y (cadr element2))
    (setq P2_z 0)
    (vla-AddLine _model_ (vlax-3D-Point P1_x P1_y
P1_z)
(vlax-3D-Point P2_x P2_y
P2_z))
    (setq i (+ i 1))
  );repeat
  (command "_zoom" "_e")
  (command "_regen")
);defun
```



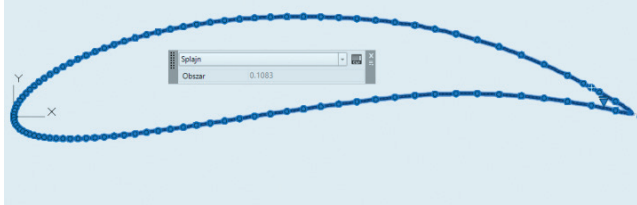
Rys. 5. Kod podprogramu wprowadzającego segmenty liniowe profilu lotniczego

```
(defun c:3() ;POLILINIA 2D
  ...
  (command "_pedit" (entlast) "t" "")
  (command "_pedit" (entlast) "d" "ws" "" "")
  (command "_zoom" "e")
  (command "_regen")
) ;defun
```



Rys. 6. Kod podprogramu wprowadzający profil lotniczy jako obiekt typu polilinia

```
(defun c:5() ;SPLAJN
  ...
  (command "_pedit" (entlast) "t" "")
  (command "_pedit" (entlast) "d" "ws" "" "")
  (command "_pedit" (entlast) "j" "")
  (command "_spline" "o" (entlast) "")
  (command "_zoom" "e")
  (command "_regen")
) ;defun
```



Rys. 8. Kod podprogramu wprowadzający profil lotniczy jako obiekt typu splajn

```
(defun c:4() ;POLILINIA 2D DOPASOWANA DO SPLAJNU
  ...
  (command "_pedit" (entlast) "t" "")
  (command "_pedit" (entlast) "d" "ws" "" "")
  (command "_pedit" (entlast) "j" "")
  (command "_zoom" "e")
  (command "_regen")
) ;defun
```



Rys. 7. Kod podprogramu wprowadzający profil lotniczy jako obiekt typu polilinia dopasowana do splajnu

- w związku z tym, że funkcja *cons* dodaje elementy do początku listy, następuje też odwrócenie danych w liście *dane_z_pliku* za pomocą funkcji *reverse*,
- następnie użytkownik ma możliwość wyboru typu wprowadzanych obiektów tworzących profil lotniczy (punkt, segment liniowy, polilinia, polilinia dopasowana do splajnu, splajn).

Wybranie przez użytkownika opcji „1” powoduje:

- zdefiniowanie symbolu graficznego odpowiedzialnego za wyświetlanie punktów (użycie funkcji *pdmode*),
- wprowadzenie w obszarze modelu systemu CAx obiektów typu punkt (użycie polecenia *vla-AddPoint* oraz pętli *repeat* wraz z funkcją *nth* wybierającą elementy z listy danych).

Wybranie przez użytkownika opcji „2” spowoduje wprowadzenie profilu lotniczego za pomocą segmentów liniowych (użycie polecenia *vla-AddLine*).

Dla podprogramów definiujących inny typ wprowadzanego obiektu profilu lotniczego (polilinia, polilinia

dopasowana do splajnu, splajn) (rys. 6–8) pominięto kod wspólny dla wymienionych programów (jest on identyczny z kodem wprowadzającym segmenty liniowe). Wykorzystuje się w tym przypadku metody zmiany obiektu na polilinię, dołączenia segmentów do polilinii, zmiany polilinii na polilinię dopasowaną do splajnu czy też obiektu typu splajn.

Podsumowanie

Przeanalizowano możliwość wykorzystania nowoutworzonego polecenia języka AutoLISP do wprowadzenia w obszarze modelu programu AutoCAD obrysu profilu lotniczego. Opracowanie takiego programu ma praktyczne znaczenie w przypadku potrzeby utworzenia modelu 3D profilu. Program pozwala także na wprowadzanie dowolnych kształtów, których współrzędne punktów węzłowych zdefiniowane są w pliku tekstowym.

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 r., materiały konferencyjne, s. 515–522.
4. Warchulski J., Warchulski M. „Program «Profil.Isp»”. Warszawa: Wojskowa Akademia Techniczna, 2017.
5. <https://kojacek.wordpress.com>.
6. <http://forum.cad.pl/autocad.html>.
7. <http://polskikursblendera.pl/ksiazka/EPUB/text/section-8-3.xhtml>.
8. <http://www.ppart.de/programming/java/profiles/NACA4.html>.
9. <http://airfoiltools.com/airfoil/naca4digit>.