

Dirbtinio intelekto užuomazgos sudėtingų judesių tyrimuose

Olegas RAMAŠAUSKAS, Artūras BIELSKIS (KU)

el. paštas: ram@gmf.ku.lt, bielskis@ip.ku.lt

Ižanga

Projektuojant naujas ar tobulinant veikiančias techniškai sudėtingas darbo mašinas (robotus, manipulatorius, automatines linijas) ir siekiant optimizuoti rezultatus, visuomet tenka palyginti žinomus faktus (įvykius, sprendimus etc) su specialistų (ekspertų) žiniomis, sukauptomis per ilgesnį veiklos (sistemos gyvavimo) laikotarpį. Turimos žinių bazės paprastai būna didelės, bet nepilnos, nesusistemintos, neretai prieštaringos, ypač informacinių ir technologinių procesų domenuose [1, 2]. Sistemos *faktai* – *žinios* – *išvados* apdorojimas įprastinėmis priemonėmis yra komplikuotas ir gali pareikalauti nemažai laiko, be to, žmogus-operatorius („aktorius“) šioje sistemoje gali nepastebėti atskirų faktų ar padaryti klaidų [3].

Galimi šios problemos sprendimo keliai: *tradiciniai*, kai į konkrečios problemos sprendimą įjungiamas didelis aukštos kvalifikacijos specialistų skaičius, kaupiant žinias, apdorojant žinomus faktus ir darant išvadas, bei *modernūs*, kai panaudojama kompiuterio savybė kaupti, lyginti ir pateikti jau apdorotą informaciją. Kompiuterinės ekspertinės sistemos ir yra ta programinės įrangos dalis, galinti visiškai arba dalinai atlikti paminėtas funkcijas. Kitaip sakant, kompiuterinės ekspertinės sistemos – tai intelektualios sistemos, skirtos patyrusių specialistų (ekspertų) elgesio modeliavimui ir imitavimui, sprendžiant specializuotus taikomuosius klausimus. Ekspertinės sistemos gali būti efektyviai pritaikytos neformaliose ar mažai formaliose veiklos srityse, kur reikalingas aukštos kvalifikacijos specialistų patyrimas (informatika, medicina, ekonomika ir pan.) [3]. Taip pat gali būti konstruojamos besimokančios ekspertinės sistemos, kurios atitiktų neuroninių tinklų modeliavimo ir mokymo principus.

Darbo tikslas

Šiame darbe sumodeliuojamas sudėtingas objekto judėjimas ir įprastinis matricinis judančio taško koordinatų užrašymas keičiamas netradiciniu simboliniu pavidalu. Gautą simbolinę judesio išraišką apdorojame tam tikslui sukurta ekspertine sistema, atrinkdami užsibrėžtus kriterijus tenkinančius sprendinius. Aprašyta ekspertinė sistema gali būti pritaikyta neformaliose ar mažai formaliose veiklos srityse, kur reikalingas aukštos kvalifikacijos specialistų patyrimas (informatika, medicina, ekonomika ir pan.) [3].

Tiriamąojo objekto žinių modelis

Bet kokio pramoninio roboto ar manipulatoriaus atliekamo darbo kokybė priklauso nuo tikslaus ir suderinto visų jo darbo organo (rankos) grandžių judesių atlikimo. Įrodyta, kad modeliuojant absoliučiai tiksliai nustatyti rankos trajektorijos bei jos baigtinių taškų vietos neįmanoma [6]. Tačiau prie pageidaujamų rezultatų priartėti pakankamai dideliu tikslumu galima daugeliu būdų: parametrinio projektavimo metodais [7, 3], neuroninių tinklų metodais [8] ir kt.

Visuotiniuose kompiuterių tinkluose diegiamos ekspertinės sistemos turi šias posistemes: globalių duomenų bazių (faktų, žinių), išvadų mašinos ir vartotojo sąsajos. Šiuo atveju, domenu vadintinas sistemos pagalba sprendžiamų problemų laukas [5]. Norėdami pavaizduoti rankos judėjimą, turime rasti priimtina kuriamai ekspertinei sistemai judesio vektorių užrašymo būdą.

Tirsime objektą, kurio dalių judėjimo tikslų išraiškų nustatymas yra sudėtingas, pavyzdžiui, roboto rankos judesio trajektoriją, veikiant inercijos, įcentrinėms, Koriolio, sunkio, apkrovos jėgoms ir pan.

Tarkime, žinome rankos tašką A koordinačių sistemoje B , kuri yra perstumta ir pasukta koordinačių sistemos C atžvilgiu (1 pav.). Ignoruodami posūkį, A padėtį sistemoje B galime užrašyti transponuotu vektoriumi:

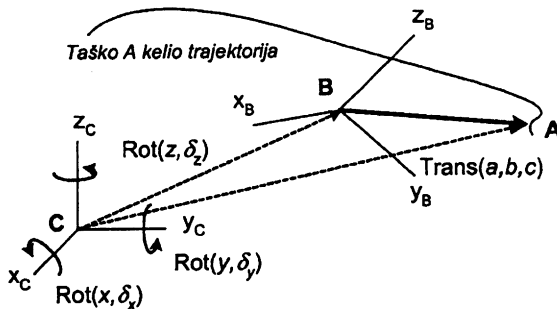
$$A = [{}^B x_A, {}^B y_A, {}^B z_A]^T, \quad (1.1)$$

čia viršutinis indeksas B išreiškia atitinkamo kintamojo priklausymą koordinačių sistemai B , kurioje jis gali būti apibrėžtas ir išmatuotas. Analogiškai, koordinačių sistemos B pradinio taško padėtį sistemoje C galime užrašyti:

$$B = [{}^C x_B, {}^C y_B, {}^C z_B]^T. \quad (1.2)$$

Norėdami susieti taško A padėtį su koordinačių sistema C , sudedame vektorius ir rezultatą užrašome eilute:

$${}^C A = {}^B A + {}^C B = [{}^C x_B + {}^B x_A, {}^C y_B + {}^B y_A, {}^C z_B + {}^B z_A]^T. \quad (1.3)$$



1 pav. Roboto rankos judesio sąrašos nustatymo diagrama.

Įvertinus posūkį ir poslinkį, roboto rankos judėjimo išraiška gali būti vaizduojama tokia eilute (žr. 1 pav.):

$${}^C_B A_{Rot} = [Trans(a, b, c)]^T \cdot [Rot(\delta)]^T, \quad (1.4)$$

arba homogeninėse koordinatėse

$${}^C_B A_{Rot} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & -\delta_z & \delta_y & 0 \\ \delta_z & 1 & -\delta_x & 0 \\ -\delta_y & \delta_x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -\delta_z & \delta_y & a \\ \delta_z & 1 & -\delta_x & b \\ -\delta_y & \delta_x & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.5)$$

Pavaizduokime matricą (1.5) sąrašine duomenų struktūra. Tam, pažymėję diferencinį judesį kiekvienos koordinatės atžvilgiu kokiu nors simboliu, kuris koduojamas pagrindinėje ASCII kodų lentelės dalyje, pavyzdžiui 65 – 90, 97 – 122, galime transformuoti judesio matricą (1.5) į paprastų simbolių eilutę:

$$A = ((1 - Z Y a)(Z 1 - Z b)(-Y X 1 c)(0 0 0 1)), \quad (1.6)$$

čia: matricos elementai – sąrašai; "(" – matricos pradžia, ")" – eilutės pabaiga, "-" – eilutės pradžia, ")" – matricos pabaiga, kai matrica skaitoma iš kairės į dešinę ir iš viršaus žemyn.

Projektuojamoji ekspertinė sistema gali atpažinti tokias sąrašinio tipo simbolių eilutes ir atlikti jų analizę. Sudarę matricas $A, B, C, \dots$ pateikiame jas ekspertinei sistemai, kaip faktus:

$$\begin{aligned} A &= ((1 - Z_{01} Y_{01} i)(Z_{01} 1 - Z_{01} j)(-Y_{01} X_{01} 1 k)(0 0 0 1)), \\ B &= ((1 - Z_{02} Y_{02} i)(Z_{02} 1 - Z_{02} j)(-Y_{02} X_{02} 1 k)(0 0 0 1)), \\ C &= ((1 - Z_{03} Y_{03} i)(Z_{03} 1 - Z_{03} j)(-Y_{03} X_{03} 1 k)(0 0 0 1)), \\ &\dots \text{visų } X, Y, Z \text{ visiems } i, j, k, \dots, \end{aligned} \quad (1.7)$$

o matricų sekas visiems a, b, c pateikiame sistemai kaip eksperto surašytas žinias:

$$\begin{aligned} A_i &= ((1 - Z_{i1} Y_{i1} a)(Z_{i1} 1 - Z_{i1} b)(-Y_{i1} X_{i1} 1 c)(0 0 0 1)), \\ C_i &= ((1 - Z_{i3} Y_{i3} a)(Z_{i3} 1 - Z_{i3} b)(-Y_{i3} X_{i3} 1 c)(0 0 0 1)), \\ &\dots, \\ K &= ((1 - Z_{03} Y_{03} i)(Z_{03} 1 - Z_{03} j)(-Y_{03} X_{03} 1 k)(0 0 0 1)); \\ A_j &= ((1 - Z_{j1} Y_{j1} a)(Z_{j1} 1 - Z_{j1} b)(-Y_{j1} X_{j1} 1 c)(0 0 0 1)), \\ B_j &= ((1 - Z_{j2} Y_{j2} a)(Z_{j2} 1 - Z_{j2} b)(-Y_{j2} X_{j2} 1 c)(0 0 0 1)), \\ &\dots, \\ L &= ((1 - Z_{02} Y_{02} i)(Z_{02} 1 - Z_{02} j)(-Y_{02} X_{02} 1 k)(0 0 0 1)); \end{aligned}$$

$$\begin{aligned}
 B_k &= ((1 - Z_{k2} Y_{k2} a)(Z_{k2} 1 - Z_{k2} b)(-Y_{k2} X_{k2} 1 c)(0 0 0 1)), \\
 C_k &= ((1 - Z_{k3} Y_{k3} a)(Z_{k3} 1 - Z_{k3} b)(-Y_{k3} X_{k3} 1 c)(0 0 0 1)), \\
 &\dots, \\
 M &= ((1 - Z_{01} Y_{01} i)(Z_{01} 1 - Z_{01} j)(-Y_{01} X_{01} 1 k)(0 0 0 1)); \\
 &\text{visų } X, Y, Z \text{ visiems } a, b, c \text{ ir } i, j, k.
 \end{aligned} \tag{1.8}$$

Matome, kad pasirinktoji užrašymo simbolika labai artima funkcinių programavimo kalbų semantikai ir todėl gali būti realizuota Visual LISP kalbos priemonėmis.

Žinoma, skaičiuojant visiškai artikuliuotos šešių laisvės laipsnių roboto rankos kinematiką išraiškos yra daug sudėtingesnės [6], pavyzdžiui:

$$P = \sum_{\substack{j=1 \\ i=1, N}}^N \left[A(i, j) \ddot{\theta}_j + B(i, j) \dot{\theta}_j^2 + \sum_{k=j+1}^N C(i, j, k) \dot{\theta}_j \dot{\theta}_k \right]; \tag{1.9}$$

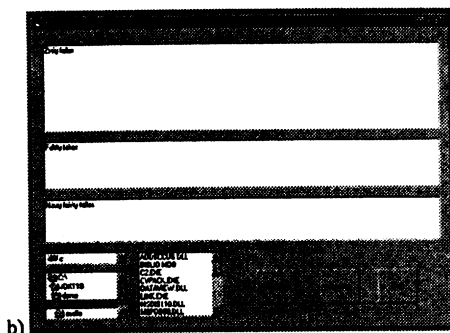
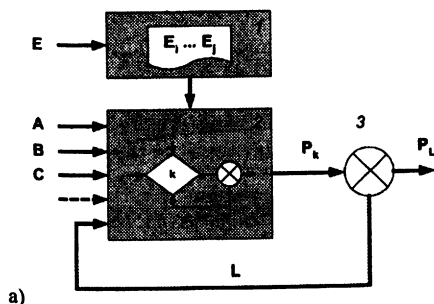
čia vien inercijos jėgas išreiškiančios matricos A elementas a_{11} gali būti sudarytas iš keliasdešimties narių, kurie, savo ruožtu, kinta priklausomai nuo posūkio θ ir kt. parametrų.

Produkcinės ekspertinės sistemos struktūrinis modelis

Darbai sukurtoji ekspertinė sistema remiasi tuo LISP kalbos ypatumu, kad joje efektyviai naudojamos rekursijos ir darbas su dideliais sąrašais. Žinių ar duomenų įterpimas į programinę Visual LISP interpretatoriaus AutoCAD terpę atliekamas interaktyviai arba naudojantis nesudėtingomis standartinėmis VBA (*Visual Basic for Application*), DCL (*Dialog Construct Language*) ar C++ priemonėmis.

Organizuojama tradicinė kartotinė struktūra (2 pav., a), kurią sudaro:

1. Eksperto žinių kaupimo modulis.
2. Tiesinio išvedimo strategijos kartotinės struktūros realizavimo modulis (išvadų mašina).
3. Faktų analizės kartotinės struktūros funkcinis modulis.



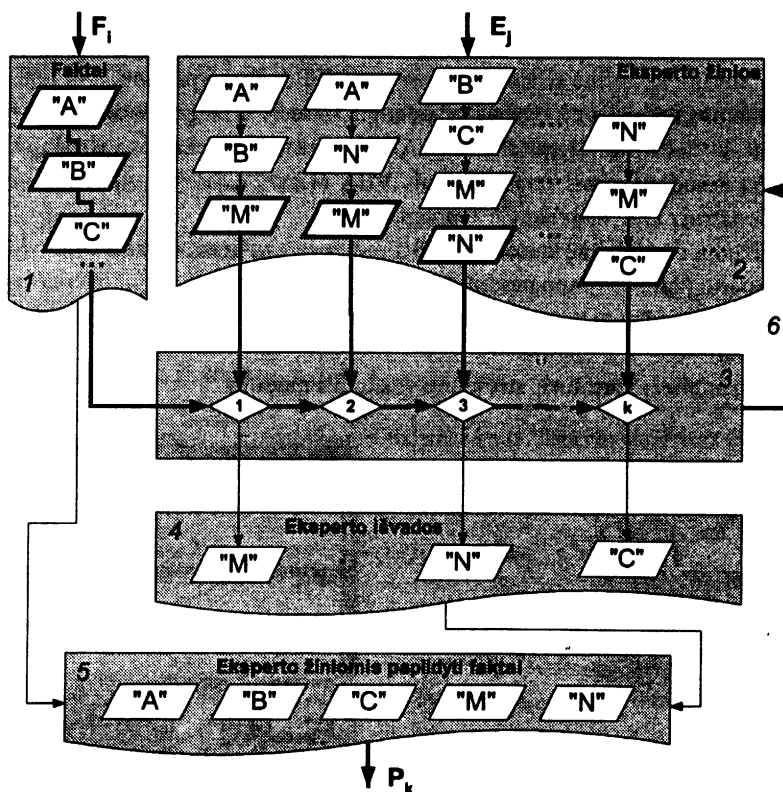
2 pav. Projektuojamos ekspertinės sistemos a) struktūrinė schema, b) pagrindinis darbo langas.

Tarkime, kad žinome faktus $A, B, C, \dots$, surašytus į tekstinį faktų failą. Be to, yra sukaupta pakankamai didelė eksperto E žinių bazė, kurioje $E_i \dots E_j$ serijomis gretinami vienareikšmiai eksperto potyriai ir žinomi faktai bei surašytos atitinkamos išvados. Išvadų mašina nuosekliai apdoroja žinių bazės įrašus ir, vadovaudamasi tiesinės išvadų strategijos principu, atrenka visas sąlygas tenkinančius faktus. Naujieji faktai nukreipiami į pradinį faktų failą ir vėl pateikiami pakartotinei ekspertizei. Procesas kartojamas, kol nebeatsiranda naujų faktų.

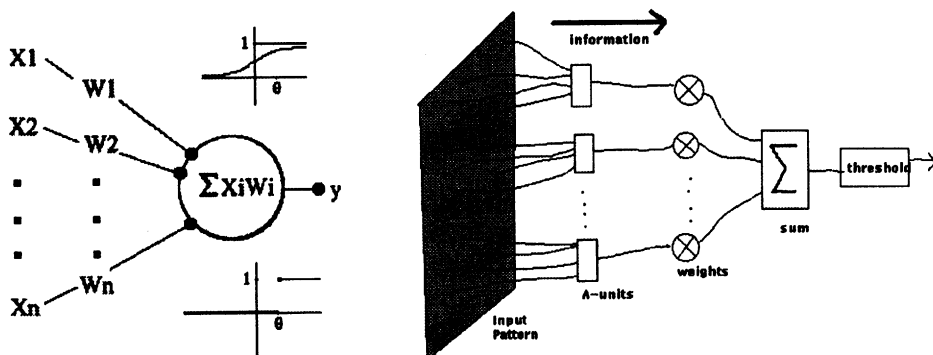
Sistemos sandara ir veikimas matyti 3 paveiksle. Globalios duomenų bazės (faktų ir žinių) bei išvadų posistemės joje realizuotos kaip atskiri funkciniai moduliai. Kadangi vartotojo sąsaja priklauso nuo to, kokioje terpėje veiks sukurtoji sistema ir gali būti realizuota įvairiomis priemonėmis, todėl schemoje neparodyta.

Atskirų funkcinį modulių paskirtis:

1. Faktų failas.
2. Eksperto žinių failas.
3. Produkcinis loginis modulis.
4. Eksperto išvadų sąrašas.
5. Eksperto išvadomis papildytas faktų failas.
6. Grįžtamoji loginė kontrolė (kartotinumumas).



3 pav. Produkcinės vieno lygio ekspertinės sistemos su tiesiniu išvedimu algoritmo blokinė schema.



4 pav. a) perceptrono skaičiuojamoji schema, b) neuroninio tinklo fragmentas [4, 8].

Nagrinėjant šios ekspertinės sistemos struktūrą, brėžinyje galima pastebėti tam tikrą panašumą su neuroninio tinklo elementų struktūra. Žinomų faktų pasikartojimas naujose žinių $E_i \dots E_j$ serijų išvadose leidžia pasikartojimo dažnio išraišką susieti su neuroninių tinklų perceptrono signalo „svorio“ koeficiento W_i išraiška (4 pav.).

Gautąsias ekspertinės sistemos išvadas interpretatorius gali pavaizduoti grafiniame ekrane kaip susietuosius objektus (roboto rankos padėties vaizdai) arba paleisti programas, galinčias CAD priemonėmis nubraižyti nustatytas sąlygas tenkinančius objektus apskaičiuotose padėtyse (parametrinio projektavimo metodai).

Duomenų gavybos technologija sistemoje

Ekspertinei sistemai pateikiami ši informacija:

1. Pirminis žinomų faktų failas (čia naudojama atsitiktinai sugeneruotų raidinių simbolių seka):

("P") ("Y") ("C") ("M")

2. Žinių bazės failas, sudarytas remiantis prielaida, kad eksperto žinios asocijuojamos su *didžiausio naudingumo* vartotojui kriterijais, įvertinus du svarbius subjektyvius *netikėtumo* ir *veiksmingumo* parametrus, o ne surašant visas įmanomas taisykles. Atsisakant nenaudingos informacijos dalies sutrumpėja rikiavimo procesai ir skaičiavimas tampa žymiai efektyvesnis [9].

Žemiau pateiktame pavyzdyje simbolių eilutės ir simbolinės konstantos vaizduojamos viena didžiąja raide. Pirmasis sąrašo narys – eilės numerio skaičius, kuris analizės procese nedalyvauja; paskutinis – patvirtintas faktas, t.y., eksperto išvada:

((1) ("A") ("B") ("C") ("D"))

((2) ("B") ("A") ("C") ("Z"))

((3) ("C") ("J"))

((4) ("P") ("A") ("R"))

((5) ("B") ("J") ("T"))

((6) ("Y") ("A"))

((7) ("P") ("E") ("H") ("A"))

((8) ("R") ("E") ("J") ("A"))

((9) ("A") ("U") ("C"))

((10) ("C") ("U"))

ir t.t.

Pirmajame ekspertizės cikle buvo aptikti "J", "A", "U" sutapimai, kurie tenkina pirminių faktų sąrašą. Papildyto faktų failo turinys:

("U") ("A") ("J") ("P") ("Y") ("C") ("M").

Po antrojo ciklo faktų failas papildomas dar vienu sutapimu "R":

("R") ("U") ("A") ("J") ("P") ("Y") ("C") ("M").

Kartojant toliau, naujų faktų neberandama, todėl duomenų gavybos (*data mining*) ciklas nutraukiamas. Sistemos veikimą suprasti ir jį valdyti padeda rezultatų registracijos log-failas. Paprastai darbo režime šis failas nespausdinamas ir neišvedamas į ekraną. Jis naudojamas sistemos darbo kontrolei ir analizės eilei nustatyti:

"... \\ES_Faktai.txt": ((P) (Y) (C) (M))

"... \\ES_Baze.txt" (žinios.):

((1) (A) (B) (C) (D))

((2) (B) (A) (C) (Z))

((3) (C) (J))

* (3) išvada-teisinga: _ J

((4) (P) (A) (R))

((5) (B) (J) (T))

((6) (Y) (A))

* (6) išvada-teisinga: _ A

((7) (P) (E) (H) (A))

((8) (R) (E) (J) (A))

((9) (A) (U) (C))

((10) (C) (U))

* (10) išvada-teisinga: _ U

...

"... \\ES_Faktai.txt": ((U) (A) (J) (P) (Y) (C) (M))

"... \\ES_Baze.txt" (žinios.):

((1) (A) (B) (C) (D))

((2) (B) (A) (C) (Z))

((3) (C) (J))

* (3) išvada-teisinga: _ J

((4) (P) (A) (R))

* (4) išvada-teisinga: _ R

((5) (B) (J) (T))

((6) (Y) (A))

* (6) išvada-teisinga: _ A

((7) (P) (E) (H) (A))

((8) (R) (E) (J) (A))

((9) (A) (U) (C))

* (9) išvada-teisinga: _C

((10) (C) (U))

ir t.t. iki atvejo, kol nebebus randama naujų faktų pasitvirtinimo.

Grafinės sąsajos ir kodo fragmentų pateiktys

Programos naudojimo palengvinimui projektuojama grafinė sąsaja (2 pav., b, parodytas vienas iš variantų, lygiavertčių sprendimų čia gali būti daug) VBA arba DCL priemonėmis. Ji naudojama žinių bei faktų skaitymui ir papildymui, o taip pat analizės tekstiniais rezultatams išvesti (5 pav.).

Ekspertinės sistemos programų failai kompiliuojami į vykdomuosius Visual LISP terpės (VLX, FAS arba ARX) modulius. Žinių bazės, faktų ir rezultatų registracijos (*log-failo*) formatai – neformatuotas tekstas (TXT).

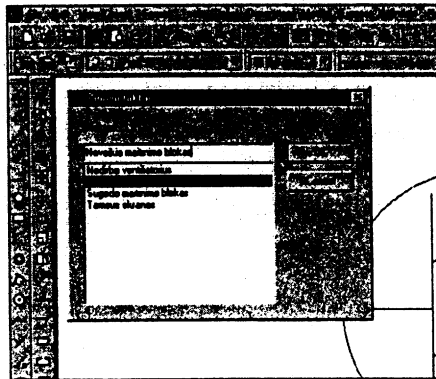
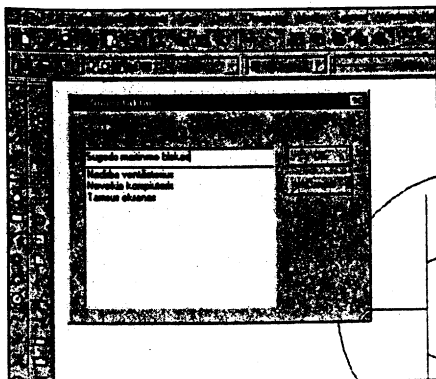
Faktų (kurie vėliau palyginami su žiniomis) failo sukūrimo arba papildymo funkcijos pavyzdys:

```
(PROGN (VL-LOAD-COM) (VL-LOAD-REACTORS) )
```

```
(SETQ *modelspace*  
  (VLA-GET-MODELSPACE  
    (VLA-GET-ACTIVEDOCUMENT  
      (VLAX-GET-ACAD-OBJECT)  
    ) ;_ Pabaiga VLA-GET-ACTIVEDOCUMENT  
  ) ;_ Pabaiga VLA-GET-MODELSPACE  
) ;_ Pabaiga SETQ, ši dalis įkraunama vienažkart
```

```
(DEFUN ff (/ ff-takas fakto-numeris faktas failas saraso-narys naujas-sa-  
rasas senas-sarapas)
```

```
(SETQ ff-takas  
(GETFILED "Faktų failas" ;|lango antraštė|; "Faktai" ;|failo vardas|;  
"txt" ;|failo prievardis|; 1 ;|atvertimas įrašui|;)  
;_pabaiga GETFILED  
fakto-numeris 0  
) ;_ Pabaiga SETQ
```



5 pav. Faktų failo apdorojimas automatizuotojo projektavimo sistemos AutoCAD lange.


```

(IF (DOS_FILEP ff-takas)
  (PROGN
    (SETQ failas (OPEN ff-takas "r")
      senas-sarasas '("") ;_Tuščias faktų sąrašas
    ) ;_ Pabaiga SETQ
    (WHILE (NOT
      (EQ (SETQ saraso-narys (READ-LINE failas))
        nil
      ) ;_ Pabaiga EQ
      ) ;_ Pabaiga NOT
      (SETQ fakto-numeris (1+ fakto-numeris)
        ;;_Pašalinami nereikalingi simboliai iš sąrašo
        saraso-narys (VL-STRING-TRIM " " saraso-narys)
        saraso-narys (VL-STRING-RIGHT-TRIM "\\ ") saraso-narys)
        saraso-narys (VL-STRING-LEFT-TRIM "\\ " saraso-narys)
        senas-sarasas (CONS saraso-narys senas-sarasas)
      ) ;_ Pabaiga SETQ
    ) ;_ Pabaiga WHILE
    (CLOSE failas)
    (SETQ senas-sarasas (CDR (REVERSE senas-sarasas)))
    (SETQ naujas-sarasas (REVERSE (CDR senas-sarasas)))
  ) ;_ Pabaiga PROGN
  (PROGN
    (SETQ fakto-numeris 1
      naujas-sarasas '("")
    ) ;_ Pabaiga SETQ
  ) ;_ Pabaiga PROGN
) ;_ Pabaiga IF
(WHILE (READ
  (SETQ faktas (DOS_COMBOLIST
    "Žinomi faktai"
    (STRCAT (ITOA fakto-numeris)
      " faktas: "
    ) ;_ Pabaiga STRCAT
    naujas-sarasas
  ) ;_ Pabaiga DOS_COMBOLIST
  ) ;_ Pabaiga SETQ
  ) ;_ Pabaiga READ
  (SETQ fakto-numeris (1+ fakto-numeris)
    naujas-sarasas (CONS faktas naujas-sarasas)
  ) ;_ Pabaiga SETQ
) ;_ Pabaiga WHILE
(SETQ naujas-sarasas (REVERSE naujas-sarasas))
(SETQ failas (OPEN ff-takas "wy"))
(FOREACH saraso-narys naujas-sarasas
  (PRINT (LIST saraso-narys) failas)
) ;_ Pabaiga FOREACH
(CLOSE failas)
(PRINC)
) ;_ Pabaiga DEFUN

```

Išvados

1. Kompiuterinės ekspertinės sistemos taikymas problemų sprendimui įvairiose dalykinėse srityse leidžia sumažinti tiek specialistų skaičių, tiek reikalavimus jų kvalifikacijai.
2. Matyt, taikant ES supaprastėja darbai, susiję su faktų ir žinių kaupimu tolimesniems problemos sprendimo etapams.

Literatūra

- [1] S. Gudas, Organizacinės veiklos integruotos sistemos informacinė architektūra, *Integruotos projektavimo sistemos*, Kaunas, Technologija, 4–16 (1999).
- [2] S. Gudas, IS kūrimas veiklos informacinės architektūros modelio pagrindu, *Integruotos projektavimo sistemos*, Technologija, 4–13 (2000).
- [3] Е. М. Кудрявцев, *AutoLISP¹ Программирование в AutoCAD, R14*. М., ДМК, 233–240 (1999).
- [4] Š. Raudys, Dirbtiniai neuroniniai tinklai, *Mokslas ir technika*, 4 (1999).
- [5] D. Baziukaitė, A. Bielskis, E. de Buhr, Consulting with expert systems, *Integruotos projektavimo sistemos*, Technologija, 47–72 (2000).
- [6] O. Ramašauskas, A. Bielskis, Roboto ($\theta - r$ manipulatoriaus) judesių dinamikos modeliavimas ir vaizdavimas, *Elektronika ir elektrotechnika*, 1(24), 71–83 (2000).
- [7] R. Manseur, Denavit-Hartenberg Parameters (ROBOT-DRAW, <http://www.ee.uwf.edu/robot-draw.htm>), Electrical Engineering, The University of West Florida (1999).
- [8] A. Rouvinen, H. Handroos, Robot positioning of a Flexible Hydraulic Manipulator Utilizing Genetic Algorithm and Neural Networks, Lappeenranta University, Laboratory of Machine Automation, (<http://www-user.lut.fi/kote/konstek/koautlab/aus.html>) (1999).
- [9] B. Liu, W. Hsu, S. Chen, Y. Ma, Analyzing the subjective interestingness of association rules, *IEEE Intelligent Systems*, 9/10, National University of Singapore, 47–55 (2000).
- [10] A.A. Bielskis, *Trimatė grafika, programavimo elementai*, Smaltija (2000), 9–31.

The application of expert system to exploration of the robot motion

O. Ramašauskas, A. Bielskis

This article expands robot motion dynamics solutions in the symbolic form and analyses the results with simple newly designed expert system (XPS). The coherency using XPS principles with artificial neural networking methods also discussed. The software implementation in Visual LISP environment examples and knowledge base files are presented.