

CAD/CAM-ის ჯგუფში დამუშავებული სახარატო ოპერაციების პროექტირების ინტეგრირებული სისტემის მიმოხილვა

დოც. ა.შარმაჯანაშვილი

წარმოების საკონსტრუქტორო და ტექნოლოგიური მომზადების ეფექტურობა მნიშვნელოვანწილადაა დამოკიდებული საკონსტრუქტორო-ტექნოლოგიური პროექტირების პროცესის ინტეგრაციაზე.

ცნობილია, რომ საკონსტრუქტორო-ტექნოლოგიური პროექტირება წარმოადგენს მრავალეტაპიან პროცესს. ამ იერარქიაში ინტეგრაციის ყველაზე დაბალი დონე - ოპერაციული ტექნოლოგიის დონეა, რომელზეც ხორციელდება ოპერაციის პროექტირებისა და პროგრამირების ამოცანების გადაწყვეტა:

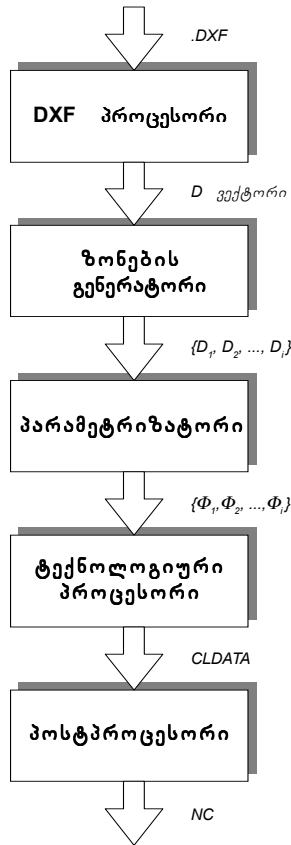
- საკონსტრუქტორო ნახაზიდან დეტალის ოპერაციული ნახაზის მიღება
- ოპერაციულ ნახაზზე მექანიკური დამუშავების ზონების გამოყოფა და ოპერაციის გეგმის სინთეზი
- ინსტრუმენტის გადაადგილების ტრაექტორიისა და ჭრის რეჟიმების სინთეზი
- მმართველი პროგრამების პოსტპროცესირება.

ოპერაციული ტექნოლოგიის დონეზე პროექტირების პროცესის ინტეგრაცია საშუალებას იძლევა:

1. ამაღლდეს საპროექტო გადაწყვეტების საიმედოობა, შემცირდეს მმართველი პროგრამების გამართვის ხანგრძლიობა და გამორიცხულ იქნას უხეში შეცდომები საწარმოო პროცესის ტექნოლოგიურ უზრუნველყოფაში.
2. შემცირდეს დრო ოპერაციული ტექნოლოგიის პროექტირებაზე და შემცირდეს შესაბამის სამუშაოთა მოცულობა.
3. გაიზარდოს საპროექტო გადაწყვეტილებების ხარისხი მათი ოპტიმიზაციის ხარჯზე.

№381 ლაბორატორიასთან არსებულ CAD/CAM-ის ჯგუფში წლების განმავლობაში მიმდინარეობდა სამუშაოები პროექტირების აღნიშნული ეტაპების ინტეგრაციაზე და შესაბამისი გამჭვლი პროექტირების არხის შექმნაზე. ამ მიმართულებით ჩატარებულმა საკვლევ-სამეცნიერო სამუშაოებმა ასახვა ჰპოვეს 6 სამაგისტრო და 1 საკანდიდატო დისერტაციებში. შეიქმნა სახარატო ოპერაციების ავტომატიზებული პროექტირების ინტეგრირებული სისტემა **Para/CAM**. პროექტირების ძირითად მეთოდოლოგიად მიღებულ იქნა პარამეტრიზებული დაპროექტების კონცეფცია.

ოპერაციული ტექნოლოგიის ზემოთ აღწერილი ამოცანების გადაწყვეტა ხორციელდება 5 ავტონომიურ ბლოკში (ნახ.1).



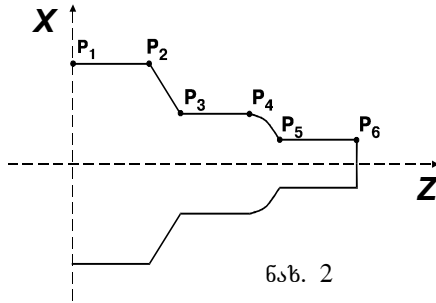
ნახ. 1

DXF პროცესორი

ბლოკის დანიშნულებაა ინტერფეისის დამყარება ავტომატიზებული კონსტრუირების სისტემასთან და საკონსტრუქტორო ნახაზიდან ოპერაციული ნახაზის პროცესირება. საკონსტრუქტორო ნახაზის წაკითხვა ხორციელდება *DXF* ფორმატში. აღნიშნული ფორმატი, რომელიც დამუშავებულია ფირმა *AutoDESK*-ის მიერ, დღეისათვის ყველაზე პოპულარულ სტანდარტს წარმოადგენს ვექტორული სახით ჩაწერილი გრაფიკული გამოსახულებების კოდირებისათვის. *DXF* ფორმატში დეტალის კონტური აღწერილია ცალკეული ელემენტარული ზედაპირებით და

განსაზღვრულია თითოეული ზედაპირის გეომეტრიული თვისებები. ბლოკის გამოსასვლელზე ოპერაციული ნახაზი მიიღება ე.წ. **D** ვექტორის სახით (ნახ.2), რომელშიც დეტალის კონტური აღწერილია საყრდენი წერტილების თანმიმდევრობით და მათი კოორდინატებით მიღებულ სისტემაში.

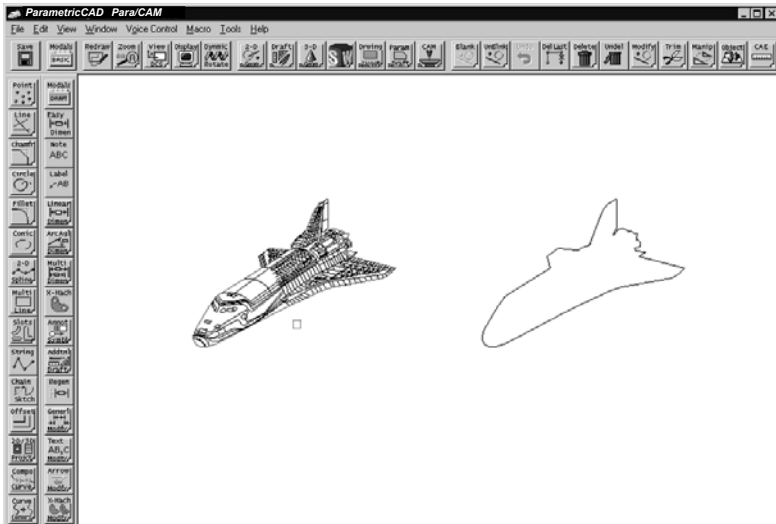
$$D=\{P_1, P_2, P_3, \dots, P_{11}\}$$



აღნიშნული ამოცანის შესრულება ბლოკში, მიმდინარეობს 4 ეტაპად:

1. *DXF* ფაილის წაკითხვა
2. კონტურის გამოყოფა
3. ზედაპირების შეუღლება
4. კოორდინატთა სისტემის შერჩევა და **D** ვექტორის გენერაცია.

კონტურის გამოყოფა ხორციელდება ინტერაქტიულ რეჟიმში მომხმარებელთან (სურ.3). ამოცანის შესრულების დროს შესაძლებელია



სურ. 3

რედაქტირების ფუნქციების - *Zoom, Select, Delete*-ს გამოყენება.

ზედაპირების შეუღლების შემოწმება ხდება ავტომატურად და საჭიროების შემთხვევაში შესაძლებელია ასევე ავტომატურად, განხორციელდეს შეუღლება.

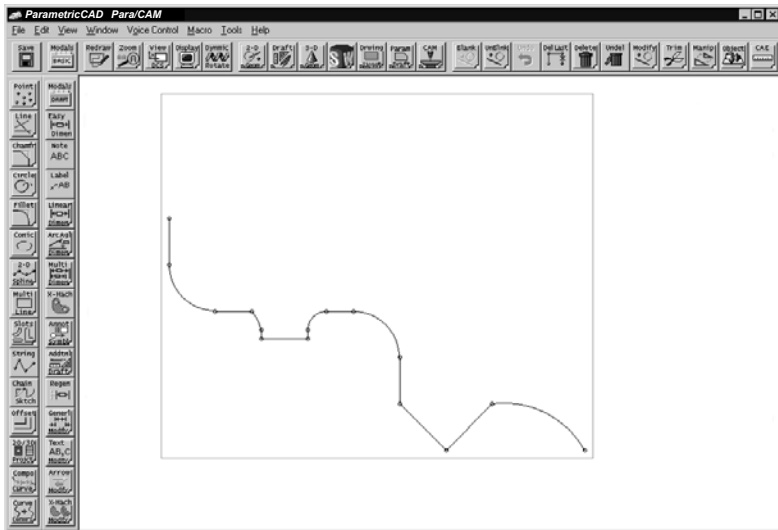
კოორდინატთა სისტემისა და **D** ვექტორის გენერაციის პროცედურაც ავტომატურია, რომელიც სრულდება მომხმარებლის მიერ კოორდინატთა სათავის შერჩევის შემდეგ.

ზონების გენერატორი

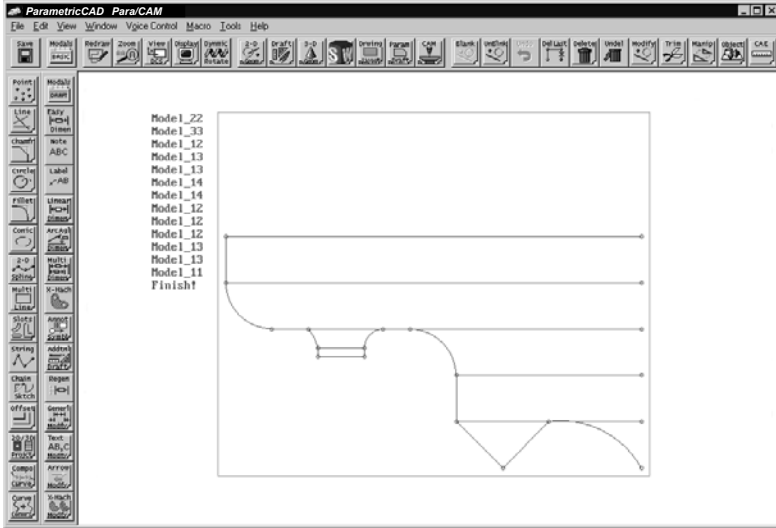
ბლოკის დანიშნულებაა დეტალის ოპერაციული ნახაზის **D** ვექტორიდან ცალკეული ზონების შესაბამისი დეტალის ფრაგმენტების **D₁, D₂, ..., D_i** გამოყოფა. პროცედურა მთლიანად ავტომატურია და ხორციელდება მხოლოდ ცილინდრული ტიპის ნამზადებისათვის (სურ. 4,5).

პარამეტრიზატორი

ბლოკის დანიშნულებაა ვექტორული სახით აღწერილი ზონის წარმოდგენა პარამეტრიზებულ ფორმაში. აღნიშნული პროცედურა ხორციელდება მონაცემთა ბაზიდან შესაბამისი პრიმიტივების იდენტიფიკაციით. პრიმიტივები მონაცემთა ბაზაში წარმოდგენილი არიან ე.წ. **D'**

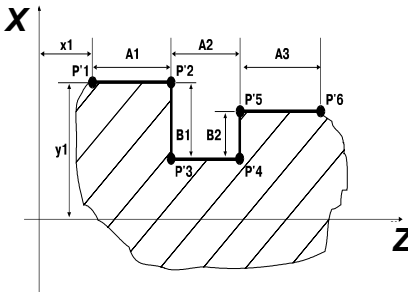


სურ. 4



სურ. 5

ვექტორით, რომლის კომპონენტები საყრდენი წერტილებია. **D** ვექტორისაგან განსხვავებით, რომლის კომპონენტებიც განსაზღვრულია რიცხვით სიმრავლეზე, **D'** ვექტორის კომპონენტები განსაზღვრულია ფორმალურ პარამეტრთა სიმრავლეზე (სურ. 6).



სურ. 6

პროცედურა სრულდება ინტერაქტიულ რეჟიმში - თითოეული ზონისთვის დგინდება შესაბამისი პრიმიტივი მონაცემთა ბაზიდან და შემდგომ ამისა, ზონის გეომეტრიული თვისებების ანალიზით ისაზღვრება ფორმალური პარამეტრების რიცხვითი მნიშვნელობები. შედეგად, მიიღება თითოეული ზონის პარამეტრიზებული აღწერა $\{\Phi_1, \Phi_2, \dots, \Phi_k\}$.

ტექნოლოგიური პროცესორი

ბლოკის დანიშნულებაა ინსტრუმენტის გადაადგილების ტრაექტორიის სინთეზი და ტექნოლოგიური ოპერაციის პარამეტრული ოპტიმიზაცია. აღნიშნული გადაწყვეტილებების მიღება ხორციელდება წინასწარ მომზადებული და აპრობირებული ტიპური გადაწყვეტების საფუძველზე. თითოეული გადაწყვეტა იმავდროულად დაკავშირებულია კონკრეტულ პარამეტრულად აღწერილ ზონასთან $\{\Phi_1, \Phi_2, \dots, \Phi_n\}$ და ინსტრუმენტთან, რომლის შერჩევაც ინტერაქტიულ რეჟიმში სრულდება. ტიპური გადაწყვეტები ცოდნის ბაზაში გაფორმებულია ტექნოლოგიური პროგრამების სახით, რომელთა შედგენისთვისაც დამუშავებულია ობიექტზე ორიენტირებული იდეოლოგია. ამ მიმართულებით დამუშავდა ობიექტების სამი კლასი - გეომეტრიული, ინსტრუმენტის გადაადგილების და პარამეტრული ოპტიმიზაციის.

გეომეტრიული ობიექტების დანიშნულებაა პარამეტრების მოცემული რიცხვითი მნიშვნელობებისათვის კონტურის გეომეტრიული გარდაქმნა. გადაადგილების ობიექტები, გარდაქმნილი კონტურისათვის ავტომატურად აფორმირებენ ტრაექტორიის საყრდენ წერტილებს. პარამეტრული ოპტიმიზაციის ობიექტები ფუნქციონირებენ ოპტიმიზაციის სტრატეგიების მიხედვით, რომელთა განსაზღვრა წარმოებს ოპერაციის სტრუქტურული სინთეზის დროს. პარამეტრული ოპტიმიზაციის სტრატეგია დამუშავებულია ტეილორის ემპირიული დამოკიდებულებისათვის და ამდენად თითოეული მათგანი წარმოადგენს ორი სასაზღვრო პირობის ერთობლიობას, რომელთა ერთობლივი ამოხსნა იძლევა ჭრის რეჟიმების (ჭრის სიჩქარე, მიწოდების სიჩქარე) ოპტიმალურ მნიშვნელობებს. სინთეზირებული ტრაექტორია და ჭრის რეჟიმების ოპტიმალური პარამეტრები ფორმდება სპეციალური *CLDATA* პროგრამის სახით.

პოსტპროცესორი

ბლოკის დანიშნულებაა *CLDATA* პროგრამიდან კონკრეტული რკმ ჩარხისათვის მმართველი პროგრამის მიღება და ასევე საჭიროების შემთხვევაში უკუ პროცესის განხორციელება. პროცედურა მთლიანად ავტომატურია. ბლოკს გააჩნია ინვარიანტული პოსტპროცესირების შესაძლებლობა. ამასთან ადაპტაცია ხორციელდება როგორც გამოსასვლელ, ასევე შესასვლელ ფორმატებზე ანუ ბლოკი ინვარიანტულია როგორც სხვადასხვა ტიპის რკმ ჩარხების მიმართ, ასევე სხვადასხვა ფორმატის *CLDATA*-ების მიმართაც. აღნიშნული თავისებურება ბლოკის ავტონომიურად გამოყენების საშუალებას იძლევა, როგორც შემათანხმებელი კვანძისა სხვადასხვა ავტომატიზებული დაპროექტების სისტემებსა და რკმ ჩარხებს შორის.

ამრიგად, აღწერილი სახარატო ოპერაციების ავტომატიზებული პროექტორების ინტეგრირებული სისტემა შეგვიძლია წარმოვიდგინოთ, როგორც შემდეგი სახის პროდუქციული სისტემა:

$$\left\{ \begin{array}{l} \Upsilon = \{ \Upsilon_1, \Upsilon_2, \Upsilon_3, \Upsilon_4, \Upsilon_5 \} \\ \Upsilon_1: DXF \rightarrow D \\ \Upsilon_2: D \rightarrow \{ D_1, D_2, \dots, D_i \} \\ \Upsilon_3: \{ D_1, D_2, \dots, D_i \} \rightarrow \{ \Phi_1, \Phi_2, \dots, \Phi_i \} \\ \Upsilon_4: \{ P_1, P_2, \dots, P_i \} \rightarrow CLDATA \\ \Upsilon_5: CLDATA \rightarrow NC \end{array} \right.$$

აღნიშნული პროდუქციების რეალიზაცია, როგორც უკვე აღინიშნა, ხორციელდება ნაწილის, ინტერაქტიულ რეჟიმში მომხმარებელთან, ხოლო ზოგის კი ავტომატურად. ქვემოთ ცხრილში, ნაჩვენებია ამოცანების განაწილება ოპერატორსა და კომპიუტერს შორის შესაბამისი პროდუქციების რეალიზაციისას.

მომხმარებელი	პროდუქცია	კომპიუტერი
<ul style="list-style-type: none"> ■ კონტურის ამორჩევა ■ რელაქტირება ■ კოორდინატთა სათავეს შერჩევა 	Υ_1	<ul style="list-style-type: none"> ■ შეუღლება ■ კოორდინატების გადაყვანა ■ D ვექტორის გენერაცია
	Υ_2	<ul style="list-style-type: none"> ■ ზონებად დაყოფა
<ul style="list-style-type: none"> ■ განზოგადებული პრიმიტივის ამორჩევა 	Υ_3	<ul style="list-style-type: none"> ■ ფორმალური პრიმიტივის განსაზღვრა ■ ზონის პარამეტრიზაცია
<ul style="list-style-type: none"> ■ ოპერაციის გეგმა ■ ინსტრუმენტის შერჩევა ■ ტიპური ვადაწყვეტის ამორჩევა ■ ობტიმიზაციის სტრატეგია 	Υ_4	<ul style="list-style-type: none"> ■ გადაადგილების ტრაექტორიის გენერაცია ■ პარამეტრული ობტიმიზაცია ■ CLDATA პროგრამა
	Υ_5	<ul style="list-style-type: none"> ■ მმართველი პროგრამის გენერაცია

**დეტალის ძრით დამუშავების სიზუსტის
აზარმოვლობის ამაღლების ერთი, არსებითად
გამოუყენებელი მოვლენის შესახებ.**

დოც. გ. შანშიაშვილი

ლითონსაჭრელ ჩარხზე დამუშავებული დეტალის ზედაპირის გეომეტრიული ფორმის და სიზუსტის გამსაზღვრელი მიმდინარე ზომა (RΔ) წარმოადგენს ჭრის პროცესითა და ძალით დატვირთული ტექნოლოგიური სისტემის—ჩარხი, სამარჯვი, იარაღი, დეტალის (ჩსიდ) ზომათა ჯაჭვის ჩამკეტ რგოლს.

აღნიშნული ჯაჭვის ფორმირება პრაქტიკულად სრულდება შინაარსით და მოქმედი ფაქტორებით განსხვავებულ ორ ეტაპზე – სისტემის ჩსიდ ზომური გაწყობის და უშუალოდ დამუშავების პროცესის რეალიზაციის ეტაპზე. ამიტომ, თუ მხედველობაში არ მივიღებთ შედარებით ნელი ტემპით ცვალებად, იარაღის გაცვებით (Y_ც) და ტემპერატურული დეფორმაციით (Y_დ) გამოწვეულ სისტემატურ ცდომილებებს, ჭრის ძალით დატვირთული სისტემის – ჩსიდ ზომათა ჯაჭვის ჩამკეტი რგოლი (RΔ) შეიძლება გაიყოს ორ ნაწილად და წარმოდგენილი იქნეს სამრგოლიანი ზომათა ჯაჭვის (1) ჩამკეტი რგოლის სახით:

$$R\Delta = C\Delta + Y\Delta \quad (1)$$

სადაც CΔ—სისტემის – ჩსიდ ზომათა ჯაჭვის ჩამკეტი რგოლია იარაღის (ან ნამზადის) მუშა სვლის დასაწყისში, მას მოკლედ სისტემის (ჩსიდ) სტატიკური გაწყობის ზომას უწოდებენ;

YΔ—ჯაჭვის(1) ჩამკეტი რგოლის (RΔ) შედარებით მცირე ნაწილია, რომელიც წარმოიქმნება ჭრის ძალის გაწონასწორების და სისტემის — ჩსიდ ელემენტების დრეკადი დეფორმაციებისა და გადაადგილებების შედეგად. აქედან გამომდინარე მას ხშირად უწოდებენ ტექნოლოგიური სისტემის ზომათა ჯაჭვის ჩამკეტ რგოლზე დრეკად გადაადგილებას.

მრავალი გამოკვლევებით დადასტურებულია, რომ დეტალების მაღალ-მწარმოებლურ რეჟიმებით (კერძოდ, ჩარხის სიმძლავრის მიხედვით დასაშვები მაქსიმალური ჭრის სიღრმეებით) დამუშავების პროცესს თან ახლავს დრეკადი გადაადგილების (YΔ), შედარებით დიდ ფარგლებში შემთხვევითი ხასიათის გადახრები. ამიტომ, აღნიშნულ რეჟიმებზე დეტალების დამუშავების პროცესში შეიძლება პირობების დაცვით

$$Y\Delta \rightarrow 0 \quad (2)$$

$$Y\Delta \rightarrow \text{const} \quad (3)$$

არა მარტო უშუალოდ და მნიშვნელოვნად იზრდება სისტემის – ჩსიდ ზომათა ჯგუფის ჩამკეტი რგოლის (RD) სიზუსტე და დეტალების დამუშავების მწარმოებლობა, არამედ არსებითად მარტივდება და პრაქტიკულად ადვილად რეალიზებადიც ხდება დეტალების პარტიის დამუშავების პროცესში ჯამური სისტემატიური ცდომილების კონტროლი და კომპენსაციაც სტატიკური გაწყობის ზომის (CD) სათანადო კორექტირებით.

დეტალების დამუშავების სიზუსტის ამალღებისათვის (2) ან (3) პირობის დაცვის განსაკუთრებულ როლსა და მნიშვნელობას ადასტურებს წარმოების პრაქტიკა . ცნობილია , რომ დამუშავების სიზუსტის ამალღების წარმოებაში ფართოდ გამოყენებულ მეთოდს საფუძვლად უდევს ნამზადის ზედაპირის მრავალჯერადი დამუშავების პრინციპი, რომელიც არსებითად გულისხმობს ჭრის ძალის და სისტემაში (ჩსიდ) დეფორმაციების , იტერაციულ მინიმიზაციას და მასასადამე, (2) ან (3) პირობის მოთხოვნილი სიზუსტის შესაბამისი მიახლოებით დაცვას .

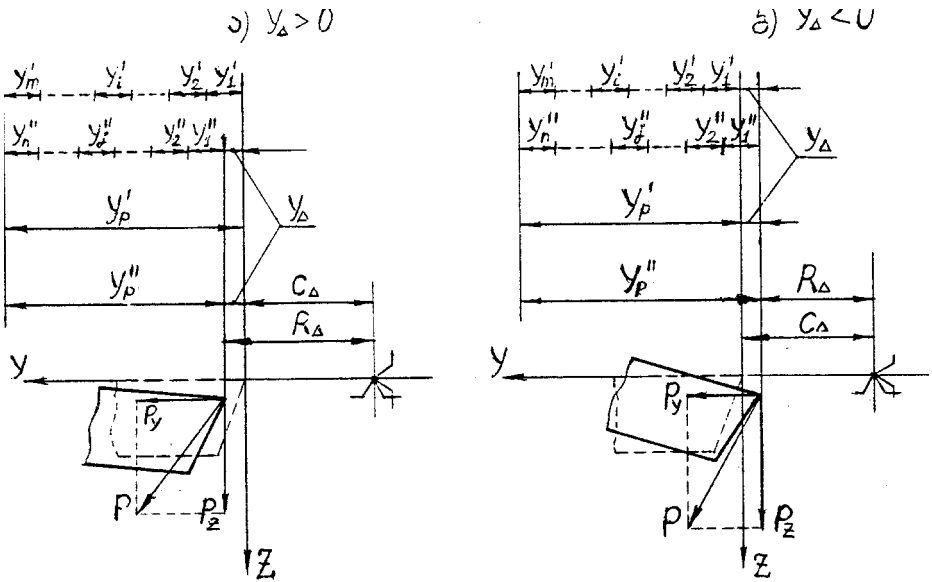
ზედაპირის მრავალჯერადი დამუშავების პრინციპის გამოყენებით დამუშავების სიზუსტის ამალღების მეთოდს გააჩნია არა მარტო ღირსებები (სიმარტივე, მაღალი საიმედოობა და სხვ.), არამედ სერიოზული ნაკლიც – სიზუსტის ამალღებას თან ახლავს დეტალების დამუშავების მწარმოებლობის შემცირება და დამუშავების თვითღირებულების მნიშვნელოვანი ზრდა.

ამ ნაკლისგან გინდაც ნაწილობრივად თავიდან აცილების დიდმა პრაქტიკულმა მნიშვნელობამ და აქტუალობამ განაპირობა ჭრთ დამუშავების სიზუსტის ამალღების სხვა გზების ძიება [1,2,3. . .] ერთ-ერთ ასეთ გზად, ზოგიერთ, შედარებით მცირერიცხოვან ნაშრომებში [4,5] მიჩნეულია დეტალების ლითონსაჭრელ ჩარხებზე დამუშავებისას ე. წ. “უსასრულო” ტექნოლოგიური სიხისტის მოვლენის გამოყენება. მაგრამ იმის გამო , რომ მკვლევართა უმეტესობა ამ მოვლენისადმი არ იჩენდნენ სათანადო ყურადღებას, დიდი ხნის განმავლობაში აღნიშნულ მოვლენას და მის პრაქტიკულ გამოყენებას ვერ მოენახა მეცნიერულად დასაბუთებული და მიმზიდველი თეორიული დამუშავება. ამ ნაკლის შევსების ცდებია გაკეთებული ნაშრომებში [6,7]. მათში, განხილული იქნა რა ჭრის ძალით დატვირთული ტექნოლოგიური სისტემა (ჩსიდ), როგორც სივრცითი, მრავალი ელემენტარული დრეკადი ქვესისტემებისაგან შედგენილი და მრავალი თავისუფლების ხარისხის მქონე დრეკადი სისტემა, თეორიულად (ბაზირების და ზომათა ჯგუფების თეორიების მეთოდებისა და ცნებების გამოყენებით) და ექსპერიმენტებით დასაბუთებული იქნა, რომ სისტემის—ჩსიდ ზომათა ჯგუფის ჩამკეტ რგოლზე დრეკადი გადაადგილება (YΔ) შეიძლება გამოსახული იქნეს ჭრის ძალაზე (P_γ) დამოკიდებული ორი ფუნქციის სხვაობით:

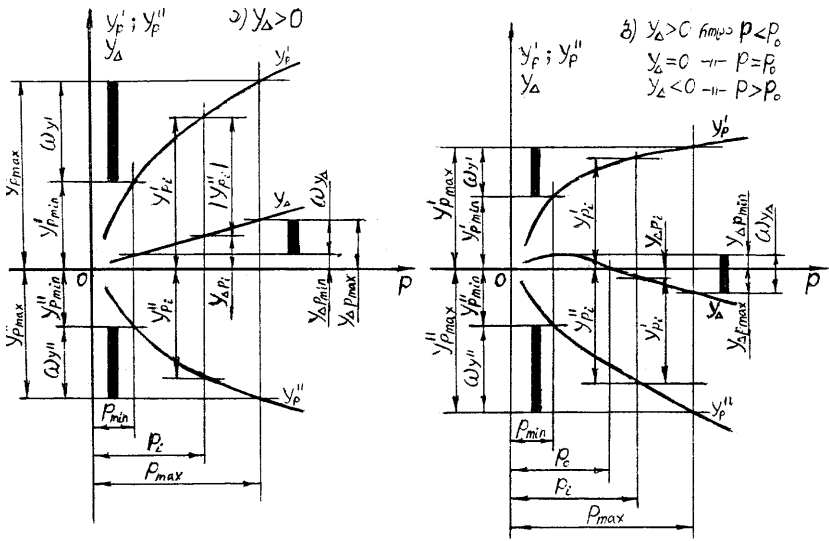
$$Y_{\Delta} =$$

სადაც Y'_i ($i=1,2,3,\dots,m$) და Y''_j ($j=1,2,3,\dots,n$) სისტემის (ჩსიდ), შესაბამისად m და n რაოდენობის, ელემენტარული დრეკადი ქვესისტემების დეფორმაციებით და გადაადგილებებით წარმოქმნილი დრეკადი გადაადგილების (Y_{Δ}) შემდგენი ელემენტარული ნაწილებია. ამთგან პირველნი ($Y'_1, Y'_2, \dots, Y'_{m_j}$) განაპირობებენ დრეკადი გადაადგილების (Y_{Δ}) ზრდას და ერთობლივობაში ქმნიან ჭრის ძალაზე დამოკიდებულ ალგებრული ფუნქციის (4) გამადიდებელ შემადგენელ ფუნქციას Y^P , ხოლო მეორენი ($Y''_1, Y''_2, \dots, Y''_n$) განაპირობებენ Y_{Δ} -ს შემცირებას და ერთობლივობაში ქმნიან იგივე ფუნქციის (4) შემამცირებელ შემადგენელ ფუნქციას — Y^P .

მე-4 გამოსახულებიდან ჩანს, რომ Y^P და Y^P ფუნქციების მნიშვნელობათა თანაფარდობისაგან დამოკიდებულებით დრეკადმა გადაადგილებამ (Y_{Δ}) შეიძლება მიიღოს დადებითი, უარყოფითი და ნულოვანი მნიშვნელობაც, ხოლო სისტემის (ჩსიდ) ტექნოლოგიურმა სისხისტემ, შესაბამისად, — დადებითი, უარყოფითი და უსასრულო მნიშვნელობა. ამასთანავე, როგორც გამოკვლევები [4,6,7] ადასტურებენ,



ნახ.1. ჭრის ძალით დატვირთული ტექნოლოგიური სისტემის (ჩსიდ) ზომათა ჯაჭვის ჩამკვეტ რგოლზე R_{Δ} დრეკად გადაადგილებათა ზომათა ჯაჭვის ფორმით მოდელირება.



ნახ.2. დრეკადი გადაადგილების Y_{Δ} და მისი გამადიდებელი და შემამცირებელი Y_p და Y_p'' ფუნქციების ჭრის ძალისაგან დამოკიდებულების გრაფიკები ორ ტიპურ შემთხვევაში

მეოფ ჩარხებზე დეტალების ჩვეულებრივი რეჟიმებით დამუშავების შემთხვევებშიც. მაშასადამე, ნახსენები მოვლენების პრაქტიკული გამოყენების საქმეში პრობლემა მდგომარეობს არა იმდენად მათ რეალიზებადობაში, რამდენადაც იმაში, თუ რა თეორიული პრინციპებისა და მეთოდების გამოყენებით შეიძლება შეიქმნას ჭრის ძალით დატვირთული ტექნოლოგიური სისტემის ზომათა ჯაჭვის ჩამკეტ რგოლზე აღნიშნული მოვლენების მიღების საიმედო პირობები და, საჭიროების შემთხვევაში, როგორიქნეს პრაქტიკულად გამოყენებული მათში შექმნილი, ან თავისთავად არსებული, ასეთი პოტენციალი.

მე - 4 გამოსახულების სტრუქტურა პარალელურ რგოლებიანი ზომათა ჯაჭვის ჩამკეტი რგოლის გამოსახულების ანალოგიურია. ამიტომ დრეკადი გადაადგილება (Y_{Δ}), რომელიც მონაწილეობს შემადგენელ რგოლად ზომათა ჯაჭვში (1), თავისმხრივ შეიძლება მოდელირებული იქნეს ნახსენებ ზომათა ჯაჭვიან იერარქიულად დაკავშირებული სხვა (მე-4 დამოკიდებულების ადეკვატური) ზომათა ჯაჭვის ჩამკეტ რგოლად და გამოსახული იქნეს იგი ზომათა ჯაჭვის სქემის ფორმითაც (იხ. ნახ.1) და ჭრის ძალაზე დამოკიდებული Y_p, Y_p'' და დრეკადი გადაადგილების (Y_{Δ}) ფუნქციების გრაფიკების ფორმითაც (იხ. ნახ. 2).

ნახაზებიდან ჩანს, რომ როდესაც $Y\Delta > 0$ განსახილველ ზომათა ჯაჭვში Y^P შემადგენელი რგოლი ასრულებს მეორე შემადგენელი რგოლის Y^P გადახრის მაკომპენსირებელი რგოლის ფუნქციას — Y^P -ის ზრდას (კლებას) თან ახლავს Y^P — ის აბსოლიტური მნიშვნელობის ზრდაც (კლებაც) და მათი სხვაობა ($Y\Delta$) ან არ იცვლება, ანდა იცვლება გაცილებით მცირე ინტერვალში შემადგენელი რგოლების ცვლილების ინტერვალებთან შედარებით. როდესაც $Y\Delta < 0$, მაკომპენსირებელი რგოლის როლში გამოდის უკვე არა შემამცირებელი (Y^P), არამედ გამადიდებელი (Y^P) რგოლი.

განსახილველ ზომათა ჯაჭვში ღრეკად გადაადგილებათა არსებითად თვითკომპენსაციის აღნიშნული მოვლენის წარმოშობა და არსებობა იმით არის გამოწვეული და უზრუნველყოფილიც, რომ განსახილველ ჯაჭვში გამადიდებელი და შემამცირებელი რგოლები და მათი გადახრები წარმოიქმნებიან, არსებობენ და იცვლიან თავიანთ მნიშვნელობას ერთი და იგივე ფაქტორის — ჭრის ძალის მოქმედებით. ამავე მიზეზით, როგორც ნახ. 2- ზე მოტანილ გრაფიკებიდან ნათლად ჩანს, ღრეკადი გადაადგილების ($Y\Delta$) არა მარტო ნებისმიერი ფიქსირებული მნიშვნელობა ($Y\Delta, P$) წარმოიქმნება Y^P , და Y^P ფუნქციათა ფიქსირებულ მნიშვნელობათა სხვაობით, არამედ ღრეკადი გადაადგილების ($Y\Delta$) გადახრათა ინტერვალის (Δy) წარმოიქმნება იგივე ფუნქციების გადახრათა ინტერვალის (Δy და $\Delta y'$) სხვაობით. ჭრის ძალით დატვირთული ტექნოლოგიური სისტემის (ჩსიდ) ზომათა ჯაჭვის ჩამკეტ რგოლზე ღრეკადი გადაადგილების ($Y\Delta$), ზომათა ჯაჭვის საშუალებით მოდელირება შესაძლებლობას იძლევა არა მარტო ფრიად მარტივად და ნათლად ახსნილი იქნეს სისტემის — ჩსიდ ე. წ. “უარყოფითი” და “უსასრულო” ტექნოლოგიური სიხისტის მოვლენები, არამედ იძლევა იმის შესაძლებლობასაც, რომ გავაკეთოთ შემდეგი დასკვნებიც:

1. დეტალების ჭრით დამუშავების პროცესს თან ახლავს ტექნოლოგიურ სისტემაში (ჩსიდ) წარმოქმნილ ღრეკად გადაადგილებათა გარკვეული ხარისხით თვითკომპენსაციის მოვლენა, რომელიც ჯერ კიდევ საკმარისად არ არის შესწავლილი, თეორიულად დამუშავებული და პრაქტიკულად გამოყენებული ჭრით დამუშავების სიზუსტისა და (ან) მწარმოებლობის ამაღლებისათვის.

2. ღრეკადი გადაადგილების ($Y\Delta$) გადახრათა ინტერვალის (Δy) ნულოვან მნიშვნელობასთან მიახლოებისათვის აუცილებელი არ არის ნამზადის ზედაპირის მრავალჯერადი დამუშავება და ჭრის ძალის ნულოვან მნიშვნელობასთან მიახლოება. არსებობს პრინციპული შესაძლებლობა ღრეკადი გადაადგილების ($Y\Delta$) გადახრათა ინტერვალის (Δy) ნულოვან მნიშვნელობასთან მიახლოება მიღწეული იქნეს ჭრის ძალის მნიშვნელოვან ფარგლებში ცვლილების პირობებშიც. ამისათვის საკმარისია

უზრუნველყოფილი იქნეს დრეკად გადაადგილებათა ზომათა ჯაჭვის გამადიდებელი (Y_P) და შემამცირებელი (Y'_P) რგოლების , როგორც ჭრის ძალის ფუნქციების პრაქტიკულად საკმარისი იდენტურობა. ამასთან დაკავშირებით ,საინტერესოა აღინიშნოს, რომ არსებობს ამ ფუნქციათა ასეთი (იდენტური) წყვილების თეორიულად უსასრულო და პრაქტიკულად საკმაოდ დიდი რაოდენობა.

3. ტექნოლოგიური სისტემის (ჩსიდ) ცალკეული ელემენტების სიხისტის გაზრდის ან შემცირების დრეკად გადაადგილებაზე (Y_Δ) გავლენის ხასიათი დამოკიდებულია იმაზე , თუ განსახილველი ტექნოლოგიური სისტემისა და დამუშავების პირობებისათვის რა ნიშანი აქვს დრეკად გადაადგილებას (Y_Δ) როდესაც Y_Δ> 0 მაშინ ფუნქციათა Y_P და Y'_P იდენტურობის ხარისხი შეიძლება გაიზარდოს და Y_Δ-ს გადახრათა ინტერვალი (Δ_Y) შემცირდეს გამადიდებელი ფუნქციის Y_P ფორმირებაში მონაწილე დრეკადი ქვესისტემების სიხისტის გაზრდით და (ან) შემამცირებელი ფუნქციის Y'_P ფორმირებაში მონაწილე დრეკადი ქვესისტემების სიხისტის შემცირებით როდესაც Y_Δ< 0 ,მაშინ გვაქვს აღნიშნულის შებრუნებული პირობები – დრეკადი გადაადგილების (Y_Δ) გადახრათა ინტერვალის (Δ_Y) შესამცირებლად საჭიროა გამადიდებელი ფუნქციის Y_P ფორმირებაში მონაწილე დრეკადი ქვესისტემების სიხისტის შემცირება და (ან)შემამცირებელი ფუნქციის Y'_P ფორმირებაში მონაწილე დრეკადი ქვესისტემების სიხისტის გაზრდა.

თუ ზემოთ მოტანილ დასკვნებთან ერთად გავითვალისწინებთ იმ გარემოებასაც , რომ მოცემული კონკრეტული (ექსპლუატაციისაში მყოფი) ტექნოლოგიური სისტემის (ჩსიდ) ზომათა ჯაჭვის ჩამკეტ რგოლზე დრეკად გადაადგილებათა ჯაჭვში გამადიდებელი Y_P და შემამცირებელი Y'_P ფუნქციების მნიშვნელობათა თანაფარდობა არსებითად არის დამოკიდებული ჭრის ძალის მიმართულებაზე , ცხადი ხდება , რომ დრეკად გადაადგილებათა თვით კომპენსაციის მოვლენის საიმედო მეცნიერული საფუძვლების შექმნისა და ეფექტიანი პრაქტიკული გამოყენებისათვის საჭიროა არსებითად ერთი პრობლემის ორი შემდეგი ამოცანის გადაწყვეტა :

ა) დრეკადი გადაადგილების (Y_Δ)გამადიდებელი Y_P და შემამცირებელი Y'_P ფუნქციების ფორმირებაში მონაწილე ტექნოლოგიური სისტემის (ჩსიდ) ყველა ელემენტარული დრეკადი ქვესისტემის სიხისტის საკუთარ მახასიათებლების განსაზღვრისათვის საიმედო თეორიისა და პრაქტიკული მეთოდის დამუშავება;

ბ) დრეკადი გადაადგილების (Y_Δ) ჭრის ძალისაგან ისეთი ანალიზური დამოკიდებულების დამყარება , რომელშიც ასახული იქნება ჭრის ძალის სიდიდისა და მიმართულების გავლენაც და ზემოთ ნახსენებ ორივე ფუნქციის (Y_P და Y'_P) ფორმირებაში მონაწილე ყველა ელემენტარული

დრეკადი ქვესისტემის სისხისტის საკუთარი მახასიათებლების გავლენაც.

ანალიზმა და გამოკვლევებმა ცხადყო, რომ ჭრის ძალით დატვირთული ტექნოლოგიური სისტემის ზომათა ჯაჭვის ჩამკეტ რგოლზე დრეკადი გადაადგილების (YΔ) წარმოქმნის მექანიზმის ზომათა ჯაჭვის ფორმით მოდელირება საშუალებას იძლევა გადაიჭრას აგრეთვე აღნიშნული ორივე ამოცანა, მაგრამ, იმის გამო, რომ ისინი იმსახურებენ სპეციალურ განხილვას და ვერ თავსდებათ წინამდებარე ნაშრომის ჩარჩოებში, აღნიშნული ამოცანების გადაწყვეტა მოხსენებული იქნება სტუ-ს პროფესორ მასწავლებელთა და (ან) მაგისტრანტთა მომავალ კონფერენციაზე .

309.1997წ

ლიტერატურა

- 1= Cfvjgjllyfcnhfbdf.obtcz cnfyrb eghfdktybz egheubvb gthtvtotybvzb cbcntvs cgb1= Gjl ht1= <= C= <fkfribyf V= @VfiByjchjtybt=_ 1970= 415c=
- 2= Flfgnbdyjt eghfdktybt cnfyrfvb= Gjl ht1= <= C= <fkfribyf V= VfiByjchjtybt=_ 1973= 684c=
- 3= *Mukthy R L Interaction of Machine Tool and Workpiece Rigidities Int I Mach Tool Desvol 10 317-325 Pecyma Press 1970 Printed in Great Britain.*
- 4= Xthybitd T= B= Bccktljdfybt htuekbhjdifybz ;tcnrjcnb njrjhysh[cnfyrd chtlyb[hfpvthjd^ rfr chtlencj gjdsitybz njxyjcnb j,hf,jnrb= DPGB 1956=
- 5= Htitnjd L= Y= Dkbzybt abpbrj-nt[ybxtcrb[afrnjhjd yf njxyjcnm j,hf,jnrb d vfiByjchjtybb= C< Jcyjdyst djghjcs njxyjcnb dpfbvjbpvtyztvjcjb b nt[ybxtcrb[bpvthtybb d vfiByjchjtybb- V= Vfiubp 1958=
- 6= Ifyibfidbkb U= L= R nt jhbb bydfhbfnyjcnb hfpvthyjb wtgb ghjwccf httpfybz= NhelS UGB #1 1967= 199-218c=
- 7= Kjkflpt N= Y= Ifyibfidbkb U= L= Ghbywogs eghfdktybz hfpvthyjb njxyjcn. ghb j,hf,jnrt ltnfktb yf vtnfkjht;eobl cnfyrf[= C< fdnjbfnbpfwbz ghjuhfvb- hjdfybz b rjlhbhjdifybt d vfiByjchjtybb= V= Yferf 1969= 134-148c=

**უნივერსალური საამწყობო სამარჯვების
კონსტრუქციის ავტომატიზირებული სინთეზის
მეთოდიკა**

ბ.გოჭუჩავა, თ.ბებეჟკორი, ღნაღარაია, ლი ცუნ-ბინი

ლითონსაჭრელი სამარჯვის პროექტირების მთელი პროცესი შეიძლება დაყვანილ იქნას სამი შედარებით ურთიერთდამოუკიდებელი ამოცანის გადაწყვეტაზე (ნახ. 1):

1. დეტალის ბაზირების სქემის დადგენა;
2. დეტალის ძალური ჩაკეტვის სქემის დადგენა;
3. სამარჯვის კონსტრუქციის სინთეზი.



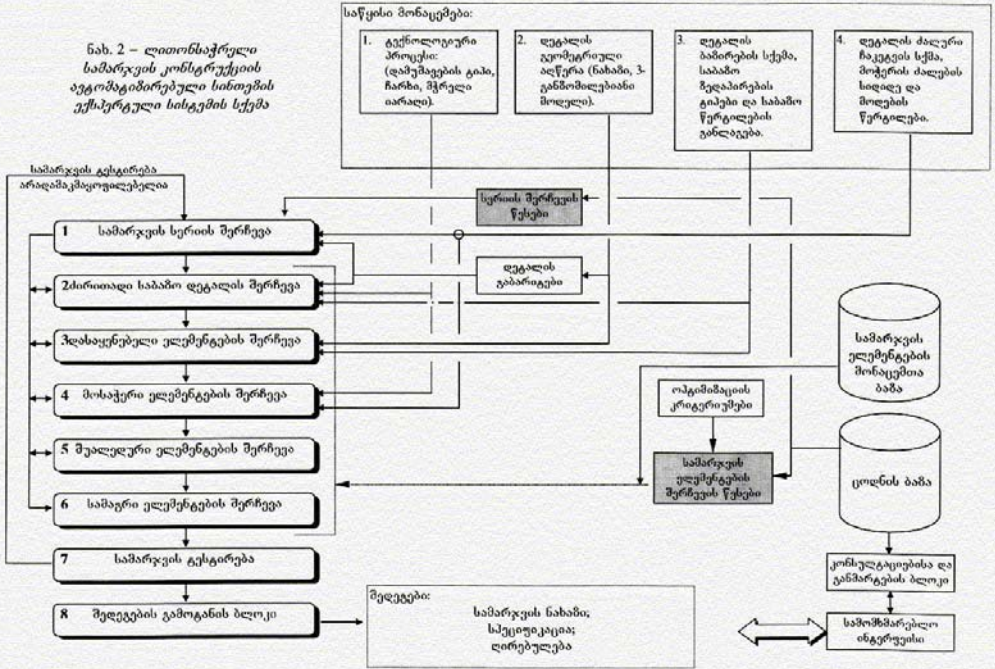
ნახ 1 – სამარჯვის ავტომატიზირებული დაპროექტების პროცესი

მოცემულ ნაშრომში წარმოადგენილია პირველი ორი ამოცანის გადაწყვეტის შედეგებზე (ბაზირებისა და ძალური ჩაკეტვის სქემები) დაყრდნობით უნივერსალური საამწყობო სამარჯვის კონსტრუქციის ავტომატიზირებული სინთეზის მეთოდიკის მიმოხილვა.

ლითონსაჭრელი სამარჯვის კონსტრუქციის ავტომატიზირებული სინთეზის ექსპერტული სისტემის სქემა მოყვანილია ნახ 2-ზე.

ექსპერტული სისტემა საწყისი მონაცემების სახით იღებს გარკვეულ მონაცემებს, რომლებიც სქემაზე მოყვანილი არიან

ნახ. 2 – ლითონსაჭრელი სამარჯვის კონსტრუქციის ავტომატიზებული სინთეზის ექსპერტული სისტემის სქემა



ბლოკში “საწყისი მონაცემები”.

ეს მონაცემები:

- 1. ტექნოლოგიური პროცესი:** დამუშავების ტიპი, ჩარხი, მჭრელი იარაღი, მისი გადაადგილების ტრაექტორია, ჭრის რეჟიმები, დამუშავების პროცესში წარმოქმნილი დატვირთვები, მოთხოვნილი სიზუსტე და სხვა;
- 2. დასამუშავებელი დეტალის გეომეტრიული აღწერა:** დეტალის ნახაზი, სამგანზომილებიანი მოდელი, ზედაპირების ტიპები, მათი დანიშნულება, ურთიერთგანლაგების სიზუსტე და სხვა
- 3. დეტალის ბაზირების სქემა:** საბაზო ზედაპირების ტიპები, მათზე განლაგებული საბაზო წერტილების რაოდენობა და განლაგების კოორდინატები და სხვა;
- 4. დეტალის ძალური ჩაქცვის სქემა:** მოჭერის ძალების მოდელის ზედაპირის ტიპები, მოჭერის ძალების სიდიდე, მოდელის წერტილების კოორდინატები, მიმართულება და სხვა.

სამარჯვის კონსტრუქციის სინთეზი ხორციელდება რვა ძირითადი, ურთიერთდამოკიდებული მოდულის საშუალებით:

1. სამარჯვის სერიის შერჩევა:

როგორც წესი უნივერსალური საამწყობო სამარჯვების კომპლექტები დაყოფილია უფრო მცირე ქვეკომპლექტებად, სერიებად (მაგ. სერიები 310, 412, 516 – Blüco-ს მაგალითზე და M8, M12, M16 - Kipp–ის მაგალითზე), რომლებიც შეიცავენ ერთი და იგივე ტიპის ელემენტებს, განსხვავებული გაბარიტებითა და ზრახნების დიამეტრებით. სამარჯვის სერიის შერჩევის ძირითადი კრიტერიუმები დასამუშავებელი დეტალის გაბარიტები და დამუშავების დროს წარმოქმნილი დატვირთვები. ამ მონაცემებსა და სერიის შერჩევის წესებზე დაყრდნობით ექსპერტული სისტემა ახდენს სამარჯვის სერიის განსაზღვრას. სხვადასხვა სერიის დეტალები არ არის ურთიერთთავსებადი, ამგვარად სერიის განსაზღვრის შემდეგ სხვა ელემენტების შესარჩევად საჭირო მონაცემთა ბაზის მოცულობა მცირდება სამჯერ, რაც აადვილებს შემდგომ მუშაობას.

2. ძირითადი საბაზისო დეტალის შერჩევა:

ძირითადი საბაზისო დეტალები – საბაზისო ფილები, კუთხედები, კუბები, წარმოადგენენ იმ ბაზას, რომელზედაც იწყობა მთელი სამარჯვის კონსტრუქცია. საბაზისო დეტალების შერჩევაზე მოქმედებს ისეთი ფაქტორები, როგორცაა დეტალის გაბარიტები, მისი დასაყენებელი ბაზისა და მჭრელი იარაღის ორიენტაცია და სხვა.

3. დასაყენებელი ელემენტების შერჩევა:

დასაყენებელი ელემენტები უზრუნველყოფენ დასამუშავებელი დეტალის ბაზირებას. მათი შერჩევა ხორციელდება დეტალის გეომეტრიის, ბაზირების სქემის, საბაზო ზედაპირების ტიპების და სხვა ანალოგიური მონაცემების მიხედვით.

4. მოსაჭერი მექანიზმების შერჩევა:

მოსაჭერი მექანიზმების დანიშნულებაა მოახდინონ დასამუშავებელი დეტალის ძალური ჩაკეტვა. მათი ტიპი დამოკიდებულია მოჭერის ძალის სიდიდეზე, მის ორიენტაციაზე, ძალის მოდების ზედაპირის ტიპზე და სხვა.

5. შუალედური ელემენტების შერჩევა:

შუალედური ელემენტები ეს ის ელემენტებია, რომლებიც აკავშირებენ სამარჯვის სხვადასხვა დეტალებს ერთმანეთთან და საბაზისო ფილასთან. შუალედური ელემენტების შერჩევა შედარებით რთულ და ძნელად

ფორმალიზებად ამოცანას წარმოადგენს, რადგან, თუ დასაყენებელი და მოსაჭერი ელემენტების შემთხვევაში არსებობს სავსებით განსაზღვრული წესები, რომლებიც იძლევა საშუალებას საკმაოდ ვიწრო ზღვრებში მოხდეს სათანადო ელემენტების ძიება, შუალედური ელემენტების შერჩევასას ასეთი წესები არაა ცხადად ჩამოყალიბებული, რაც ქმნის ერთგვარი არჩევანის თავისუფლებას, ხოლო ეს უკანასკნელი ქმნის გარკვეულ პრობლემებს ამოცანის ფორმალიზაციას და მისი რეალიზაციას. შუალედური ელემენტების შერჩევის ბლოკში ყველაზე უფრო ფართოდ გამოიყენება ექსპერტული სისტემის “ინტელექტუალური” შესაძლებლობები.

6. სამაგრი ელემენტების შერჩევა:

სამაგრი ელემენტები – ხრახნები, ქანჩები, საყელურები და სხვა, განკუთვნილი არიან სამარჯვის სხვადასხვა დეტალების ერთმანეთთან დასამაგრებლად. ამ ეტაპზე პრაქტიკულად უკვე შექმნილია სამარჯვის კონსტრუქცია, ცნობილია მისი ცალკეული ელემენტების ურთიერთგანლაგება სივრცეში და დამოკიდებულება სხვა ელემენტებთან, სწორედ ეს მონაცემები მოქმედებენ სამაგრი ელემენტების შერჩევაზე. გარდა ამისა ამ უკანასკნელთა სტანდარტულობა, მცირე არჩევანი და ის ფაქტი, რომ სამარჯვის ყველა ელემენტი პრაქტიკულად ერთნაირი პრინციპით უკავშირდება ერთმანეთს, საგრძნობლად აადვილებს ამ ამოცანის გადაწყვეტას.

7. სამარჯვის ტესტირება:

წინა ეტაპი ეტაპი პრაქტიკულად ახორციელებს სამარჯვის კონსტრუქციის სინთეზს. თუმცა დაპროექტების პროცესი ამით არ მთავრდება, რადგან აუცილებელია შემოწმდეს, შესძლებს თუ არა დაპროექტებული სამარჯვი მუშაობას რეალურ პირობებში. სამარჯვის ტესტირებისას უნდა შემოწმდეს ორი ფაქტორი: შესაბამება თუ არა სამარჯვის კონსტრუქცია გეომეტრიულ მოთხოვნებს (შესაძლებელია თუ არა მასში დეტალის ჩაყენება, ხომ არ უშლის მისი რომელიმე ელემენტი მჭრელი იარაღის გადაადგილებას ხელს და სხვა), და რამდენად უძლებს დაპროექტებული სამარჯვი დამუშავების პროცესში წარმოქმნილ დატვირთვებს (ჭრისა და მოჭერის ძალებს). თუ სამარჯვის ტესტირებისას აღმოჩნდება რაიმე გეომეტრიული ან დინამიკური უთანხმოებანი, მაშინ სისტემა იტერაციულად დაუბრუნდება პროექტირების ბლოკებსა და შეეცდება გამოასწოროს შეცდომები. შემდგომ განხორციელდება ახალი ტესტირება და ასე, იმოქმედებს სისტემა მანამ, სანამ არ მიიღებს სრულყოფილ სამარჯვის კონსტრუქციას.

8. შედეგების გამოტანა:

ბოლო ეტაპი პროექტირების პროცესში ეს არის შედეგების გამოტანა. ამ დროს ხდება დაპროექტებული სამარჯვის ნახაზებისა და სამგანზომილებიანი მოდელების შექმნა, ტექნიკური დოკუმენტაციის გაფორმება, თვითღირებულების ანგარიში და სხვა. ყოველივე ეს შედეგების სახით მიეწოდება მომხმარებელს, როგორც პასუხი მის მიერ მანქანისათვის გადაწოდებულ საწყის მონაცემებზე.

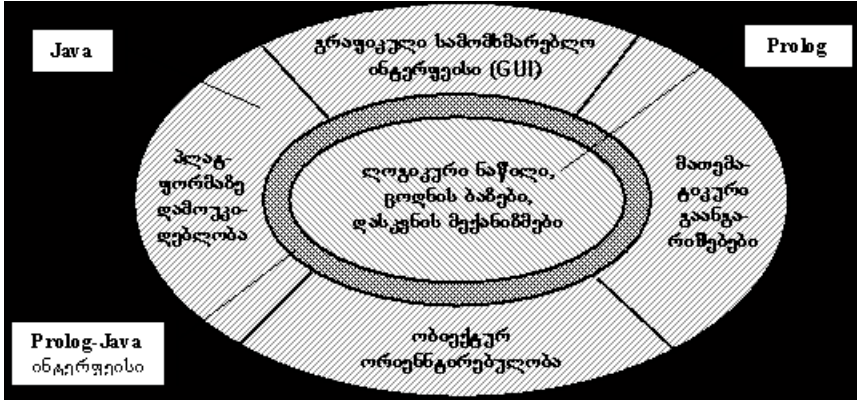
გარდა ზემოთგანხილული ბლოკებისა ექსპერტულ სისტემაში ასევე შედიან მონაცემთა და ცოდნის ბაზები.

მონაცემთა ბაზები არის ის “წყარო”, საიდანაც სისტემა ახორციელებს მონაცემების, მოცემულ შემთხვევაში – სამარჯვის ელემენტების შერჩევას. ცოდნის ბაზები წარმოადგენენ ექსპერტული სისტემის “ტვინს”, მის ინტელექტუალურ შესაძლებლობებს. აქ ფაქტებისა და “თუ-მაშინ” წესების სახით შეკრებილია ინფორმაცია იმის შესახებ, თუ სისტემამ როგორ უნდა გააკეთოს გარკვეული დასკვნები და მიიღოს გადაწყვეტილებები.

ექსპერტული სისტემებისათვის ერთერთი დამახასიათებელი ბლოკია კონსულტაციებისა და განმარტებების ბლოკი. იმის გამო რომ შეუძლებელია პროექტირების მთელი პროცესის სრული ავტომატიზაცია, აუცილებელი ხდება მუშაობის წარმართვა დიალოგურ რეჟიმში, ანუ მომხმარებელთან დიალოგში. სწორედ ამ დიალოგის პროცესში სისტემამ გარკვეული კონსულტაციები უნდა გაუწიოს მომხმარებელს და ასევე მომხმარებლის სურვილისამებრ, უნდა მისცეს მას გარკვეული განმარტებები, თუ რის საფუძველზე აკეთებს სისტემა ამათუიმ დასკვნებსა და იღებს გადაწყვეტილებებს.

და ბოლოს ამ ბლოკთან და ასევე სისტემის ყველა სხვა ძირითად მოდულებთან მჭიდროდაა დაკავშირებული სამომხმარებლო ინტერფეისის ბლოკი, რომელიც უზრუნველყოფს ურთიერთობას ადამიანსა და მანქანას შორის და ასევე ახორციელებს მუშაობის დიალოგურ რეჟიმს.

იმის გამო რომ სამარჯვის დაპროექტების ამოცანა განეკუთვნება ე. წ. “ინტელექტუალური” ამოცანების რიგს, ანუ ძნელად ექვემდებარება ფორმალიზაციასა და მით უმეტეს ალგორითმიზაციას, ზემოთ აღწერილი სისტემის რეალიზაციისათვის აუცილებელი ხდება ლოგიკური პროგრამირების გამოყენება. ლოგიკური, “აღწერილობითი” პროგრამირების, ფაქტებითა და წესებით, ინფორმაციის პრედიკატული სახით წარმოდგენის საუკეთესო საშუალებაა პროგრამირების ენა პროლოგი (**Prolog**). ამ ენაზე შეიქმნება ცოდნის ბაზები.



ნახ. 3 – ექსპერტული სისტემის რეალიზაციის სქემა

თუმცა ექსპერტულ სისტემაში გარდა ლოგიკური ნაწილისა იქნება გადასაწყვეტი ისეთი საკითხებიც, რომლებიც დაკავშირებული იქნებიან მათემატიკურ გამოთვლებთან, მონაცემთა ბაზებთან მუშაობასთან, ცოდნის ობიექტურ-ორიენტირებული სახით წარმოდგენასთან, სხვადასხვა დეტალების გრაფიკულ გამოსახვასთან და სხვა. გარდა ამისა სისტემას აუცილებლად უნდა ჰქონდეს სამომხმარებლო ინტერფეისი, იმ საშუალებების ერთობლიობა, რომლითაც მოხდება ინფორმაციის მიმოცვლა ადამიანსა და მანქანას შორის. ასეთი მოდულების დასაწერად საუკეთესო საშუალებაა ობიექტურ-ორიენტირებული პროგრამირების ენები, მაგალითად ისეთი, როგორიცაა ჯავა (Java™). ამ უკანასკნელს გააჩნია მძლავრი საშუალებები ობიექტურ-ორიენტირებული პროგრამირებისათვის, მათემატიკური გაანგარიშებებისათვის, გრაფიკული სამომხმარებლო ინტერფეისის შექმნისათვის, მონაცემთა ბაზებთან მუშაობისათვის და სხვა. ამგვარად დაგეგმილი ლითონსაჭრელი სამარჯვების ავტომატიზირებული დაბრუნების ექსპერტული სისტემა შეიქმნება ორი პროგრამირების ენის პროლოგისა და ჯავას სინთეზის საფუძველზე (ნახ. 3). ლოგიკური, “ინტელექტუალური” ნაწილი განხორციელდება პროლოგზე, ხოლო გრაფიკული სამომხმარებლო ინტერფეისი, და სხვა მსგავსი ნაწილები – ჯავაზე. ურთიერთობა ამ ორ ენას შორის განხორციელდება სპეციალური საშუალებებით.

ზომათა ჯაჭვის გადაწყვეტის ავტომატიზირებული ძვენიშნა

თ.ბებეჟკორი, დ.ნადარაია, ო.პიტავა

აწეობის პროცესის დაპროექტება ნაკეთის დამზადების საერთო ციკლის ერთ-ერთი მნიშვნელოვანი ეტაპია. ამ პროცესის განხორციელებისას ხდება ისეთი მნიშვნელოვანი ამოცანის გადაჭრა, როგორცაა მთლიანად ნაკეთისა და მისი შემადგენელი ნაწილების-კვანძების, ქვეკვანძების, კომპლექტების და დეტალების ზომათა ჯაჭვების გადაწყვეტა და შემადგენელი რგოლების სიზუსტის ქვალიტეტის და სიმქისის პარამეტრების განსაზღვრა. ამოცანის ოპტიმალურ გადაწყვეტაზე მნიშვნელოვნად არის დამოკიდებული, როგორც ნაკეთის აწეობის, ასევე მისი ცალკეული დეტალების დამზადების ტექნოლოგიური პროცესის შრომატევადობა და თვითღირებულება.

ზომათა ჯაჭვის ცნობილი ხუთი მეთოდით გადაწყვეტა (სრული ურთიერთშეცვლადობის, არასრული ურთიერთ-შეცვლადობის, ჯგუფური ურთიერთშეცვლადობის, მორგებისა და რეგულირების მეთოდებით) და შემადგენელი რგოლების სიზუსტის ქვალიტეტის განსაზღვრა ფრიად რთული, შრომატევადი ამოცანაა და ძირითადად ხორციელდება კონსტრუქტორის შრომის ხარჯზე, რაც არ გამოირიცხავს გარკვეულ ცდომილებებს ამოცანის გადაწყვეტისას. ამიტომ კომპიუტერული ტექნიკის გამოყენება მნიშვნელოვნად აადვილებს ზემოაღნიშნული ამოცანის გადაწყვეტას.

ამ მიზნით შეიქმნა ზომათა ჯაჭვის გადაწყვეტის ავტომატიზირებული ქვესისტემა **DL-CAD®** დ.ნადარაია, მისი განხორციელების ალგორითმი (იხ. ნახ. 1), დაწერილია პროგრამა პროგრამირების ენაზე **Visual Basic** და შესაბამისი მონაცემთა ბაზები.

ალგორითმის მუშაობის საწყისი მონაცემების ნუსხა დამოკიდებულია ზომათა ჯაჭვის გადაწყვეტის მეთოდსა და მეთოდის განხორციელების წესზე (ერთნაირი და სხვადასხვა ქვალიტეტის წესი).

საწყისი მონაცემები ზომათა ჯაჭვის სრული ურთიერთშეცვლადობის მეთოდის ერთნაირი ქვალიტეტის წესით გადაწყვეტისას შემდეგია:

1. წარმოების ტიპი;
2. ზომათა ჯაჭვში შემავალი რგოლების რაოდენობა;
3. მოთხოვნილი ტექნიკური პირობა მმ;
4. რგოლების ნომინალური ზომა მმ;

5. რგოლების ტიპი (გამადიდებელი თუ შემამცირებელი).

ერთნაირი ქვალიტეტის წესის გამოყენებისას ცნობილი არ არის არც ერთი რგოლის ქვალიტეტი, ყველა რგოლი მზადდება ერთნაირი ქვალიტეტით, რომელიც იანგარიშება ფორმულით

$$a_{\text{საშ}} = TA_0 / \sum_{i=1}^n (0.45\sqrt{A_i} + 0.001A_i)$$

სადაც TA_0 მოთხოვნილი ტექნიკური პირობაა, A_i რგოლების ნომინალური ზომა, $a_{\text{საშ}}$ დაშვების ერთეულთა საშუალო რიცხვია. $a_{\text{საშ}}$ -ს მიხედვით მონაცემთა ბაზიდან შეირჩევა უახლოესი უმცირესი ქვალიტეტი, რომელიც განსაზღვრავს შემადგენელი რგოლების ნომინალური ზომის დაშვებებს და შესაბამისად სიმქისის პარამეტრის მნიშვნელობას მკმ-ში.

საწყისი მონაცემები ზომათა ჯაჭვის სრული ურთიერთშეცვლადობის მეთოდის სხვადასხვა ქვალიტეტის წესით გადაწყვეტისას შემდეგია:

1. წარმოების ტიპი,
2. ზომათა ჯაჭვში შემავალი რგოლების რაოდენობა;
3. მოთხოვნილი ტექნიკური პირობა მმ;
4. რგოლების ნომინალური ზომა მმ;
5. წინასწარ ცნობილი ქვალიტეტის მქონე რგოლების ზედა და ქვედა გადახრა მმ;
6. რგოლის ტიპი.

ტექნიკური პირობა ერთნაირი ქვალიტეტის მქონე რგოლებისათვის

$$TA_0' = TA_0 - \sum_{i=1}^p TA_i = \sum_{i=1}^{n-p} TA_i$$

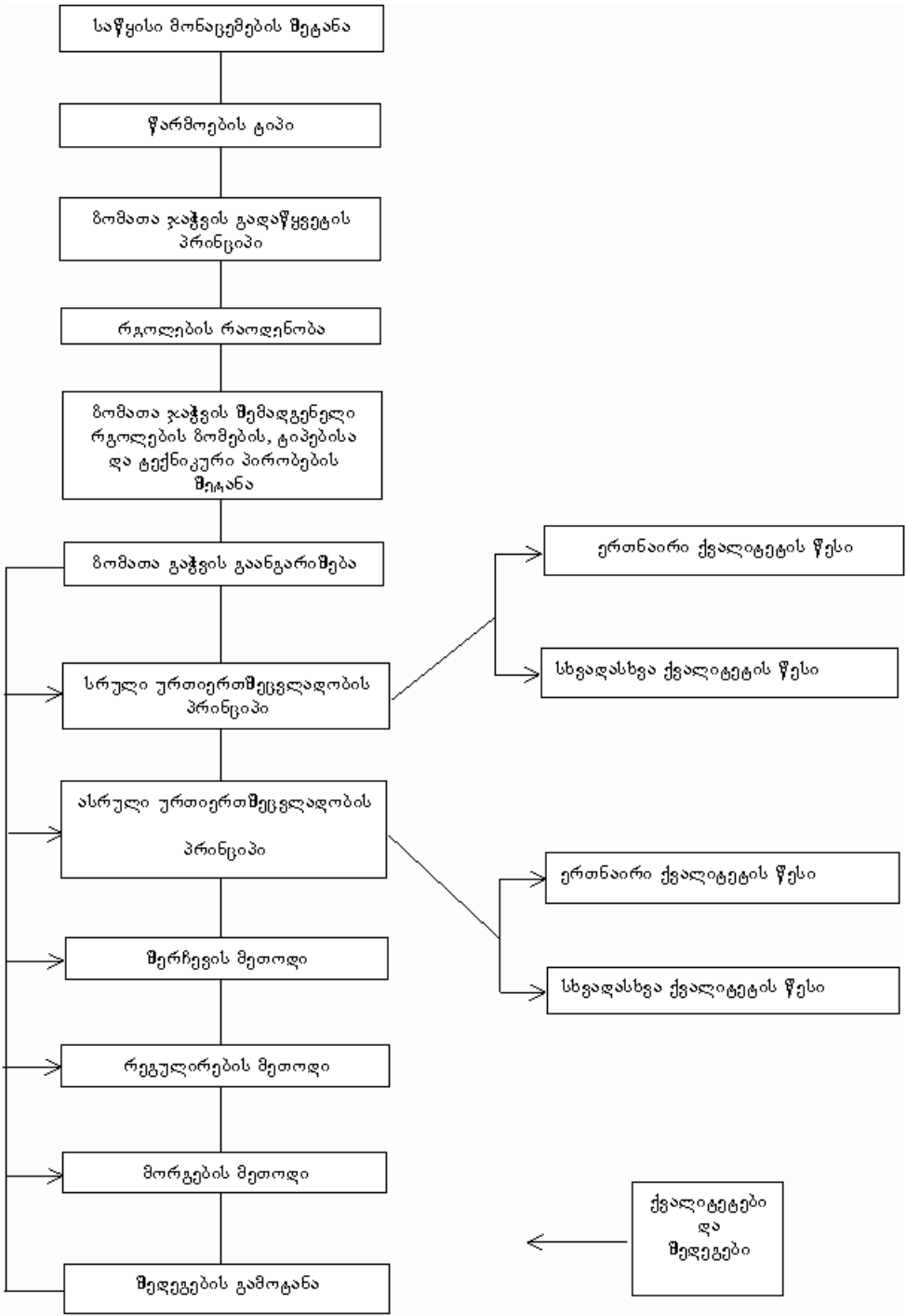
დანარჩენი რგოლების ქვალიტეტი იანგარიშება ფორმულით

$$a_{\text{საშ}} = TA_0' / \sum_{i=1}^{n-p} (0.45\sqrt{A_i} + 0.001A_i)$$

სადაც P ცნობილი ქვალიტეტის მქონე რგოლების რაოდენობაა. ქვალიტეტის და სიმქისის შერჩევა ერთნაირი ქვალიტეტის წესის გამოყენების ანალოგიურია.

არასრული ურთიერთშეცვლადობის მეთოდი მოიცავს ერთნაირი და სხვადასხვა ქვალიტეტის წესს. ზომათა ჯაჭვების ამ მეთოდით გადაწყვეტისას გაითვალისწინება განაწილების კანონი, რომელიც ხასიათდება კოეფიციენტით, ამიტომ მისი მნიშვნელობა შედის საწყისი მონაცემებში.

საწყისი მონაცემები, ზემოაღნიშნული ორივე წესის



შემთხვევაში, სრული ურთიერთშეცვლადობის მეთოდის ანალოგიურია.

ერთნაირი ქვალიტეტით ანგარიშისას

სხვადასხვა ქვალიტეტით ანგარიშისას

$$a_{\text{საშ}} = TA'_0 K_0 / \sqrt{\sum_{i=1}^n (0.45\sqrt{A_i} + 0.001A_i)^2 K_i^2}$$

$a_{\text{საშ}}$ -ს შერჩევა წინა მეთოდის ანალოგიურია.

ზომათა ჯაჭვის ჯგუფური ურთიერთშეცვლადობის მეთოდით გადაწყვეტისას საწყისი მონაცემები შემდეგია:

1. წარმოების ტიპი;
2. რგოლების რაოდენობა;
3. მოთხოვნილი ტექნიკური პირობა მმ;
4. რგოლების ნომინალური ზომა მმ;
5. რგოლების ტიპი;
6. წინასწარ მოცემული რგოლის ზედა და ქვედა გადახრა მმ.

წარმოების ტიპის გათვალისწინებით წინასწარ შერჩევა

$$a_{\text{საშ}} = TA_0 K_0 / \sqrt{\sum_{i=1}^n (0.45\sqrt{A_i} + 0.001A_i)^2 K_i^2}$$

$$n = \frac{TA'_{\text{საშ}}}{TA_0 - \sum_{i=1}^n TA_i}$$

სადაც P წინასწარ ცნობილი ქვალიტეტის მქონე რგოლის რაოდენობაა, TA_i წარმოების ტიპის მიხედვით შერჩეული რგოლების დაშვებაა. TA_0 მოთხოვნილი ტექნიკური პირობა.

გაანგარიშებით მიღებული ჯგუფების რაოდენობა

მრვალდება მთელ რიცხვამდე მეტობით ე. ი. $TA'_{\text{საშ}} = n \cdot TA_0$

ამ დროს დაცული უნდა იყოს ტოლობა

$$\sum_{i=1}^n TA_i = \sum_{i=1}^n TA_i$$

ზომათა ჯაჭვების მორგების და რეგულირების მეთოდით გადაწყვეტისას საწყისი მონაცემები შემდეგია:

1. წარმოების ტიპი;
2. მოთხოვნილი ტექნიკური პირობა მმ;
3. რგოლების ზომა მმ;

4. რგოლების რაოდენობა;
5. საკომპენსაციო რგოლი;
6. რგოლის ტიპი;
7. წინასწარ ცნობილი რგოლის ზედა და ქვედა გადახრა.

წინასწარ შეირჩევა შემაღლებულ რგოლის ქვალიტეტი და საკომპენსაციო რგოლი გაიანგარიშება. კომპენსაციის სიდიდე

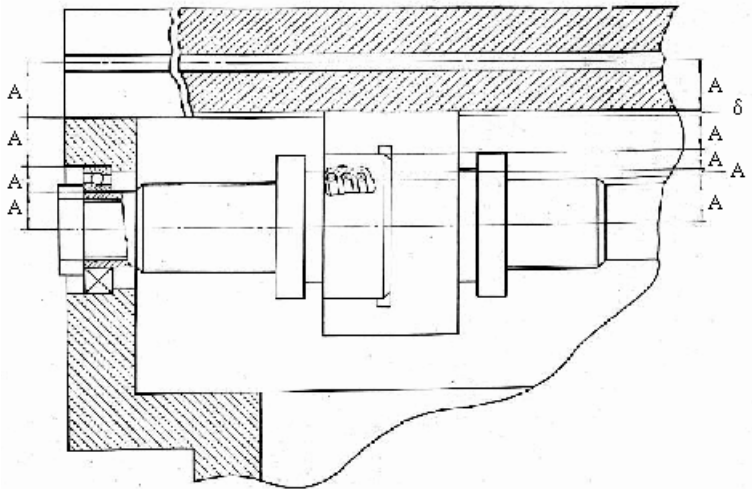
$$TA_{\text{კომ}} = TA'_0 - TA_0$$

სადაც:

$$TA'_0 = \sum_{i=1}^{k-1} TA_i$$

საწყისი მონაცემების კომპიუტერში შეყვანისა და პროგრამის განხორციელების შემდეგ, გამოსავალზე მიღებული იქნება რგოლების ქვალიტეტი და ქვალიტეტის შესაბამისი სიმქისე ზომათა ჯაჭვის გადაწყვეტის ნებისმიერი მეთოდისათვის.

ქვემოთ მოყვანილია ავტომატიზირებული სისტემის გამოყენების კონკრეტული მაგალითი ზომათა ჯაჭვისათვის (იხ. ნახ. 2) და გამოსავალზე მიღებული შედეგი (ცხრილი1)



ლიტერატურა

1. Балакшин Б.С., Основы технологии, Машиностроение М.1985.
2. Якушев А.И., Воронцов А.Н., Федосов Н.М., Взаимозаменяемость, стандартизация и технические измерения, М.1987.
3. გ. ბოკუჩავა, თ. გეგეჭკორი, ო. პიტავა. ნაკეთის აწყობის ტექნოლოგიური პროცესის ოპტიმიზაცია, სტუ შრ-ბი, №1 (412) 1997წ.

ცხრილი 1 - სრული ურთიერთშეცვლადობის მეთოდი

ნომინალური ზომა მმ.	შეღა გაღანრა მკმ	ქვეღა გაღანრა მკმ	ქვალიტეტი IT	სიმქისე Ra მკმ
A ₁ = 6	1.5	0	h2	0.2
A ₂ = 8	1.5	0	h4	0.4
A ₃ = 9.5	2	0	h2	0.2
A ₄ = 10.1	2	0	h2	0.2
A ₅ = 10	2	0	h2	0.2
A ₆ = 8.5	1.5	0	h2	0.2
A ₇ = 2.85	1.2	0	h2	0.2
A ₈ = 1.625	1.2	0	h2	0.2
A ₉ = 8	1.5	0	h2	0.2

Исследование объектно-ориентированной архитектуры САПР-ТП

аспирант Л Мегрелишвили

Традиционными задачами проектирования являются задачи синтеза и анализа. При этом под синтезом понимается процесс принятия решения, а под анализом процесс оценки принятого решения.

САПР (Система автоматизированного проектирования) - это человеко-машинная система, которая решает задачи автоматизированного синтеза и автоматизированного анализа. В процессе взаимодействия со средствами САПР в интерактивном режиме проектировщик осуществляет получение проектного решения на основе моделей синтеза и анализа. Модель - это формализованное описание знаний, при помощи которых система осуществляет интерпретацию и формирование решений.

Термин САПР является смысловым аналогом - эквивалентом английского САК (сокращено Computer Aided Resinged) и означает проектирование с помощью компьютера, человеку в этом процессе отводится активная роль.

САПР - состоит из комплекса средств автоматизированного проектирования (КСАП), взаимосвязанного с необходимыми подразделениями проектной организации.

САПР - это система проектирования, в которой органично объединены творческие усилия коллектива проектировщиков и возможности математических методов и компьютерных систем на всей совокупности взаимосвязанных этапов проектирования с применением развитых средств программного и информационного обеспечения для коренного улучшения качества проектных работ и сокращения их сроков, что достигается благодаря :

- систематизации и совершенствованию проектного процесса, которые сопровождаются перестройкой структуры и кадрового состава проектных организаций;
- применению эффективных математических моделей проектируемых объектов;
- комплексной оптимизации принимаемых решений;
- улучшению информационного обеспечения разработок;
- автоматизации трудоемких рутинных работ;
- частичной замене макетирования.

Можно выделить два основных направления исследований в разработке

САПР:

1. Изучение методологии автоматизированного проектирования ;
2. Развитие новых методов организации и разработки самих САПР .

Первое направление связано с изучением теоретического, аксиоматического и алгоритмического аспектов САПР, т.е. с разработкой технологии проектирования как науки, второе - с разработкой теоретических и практических методов реализации самих систем в рамках уже известных технологических подходов, достаточно хорошо зарекомендовавших себя в условиях промышленного применения.

Одними из основных принципов , на которых основаны промышленные САПР это:

Концептуальность .

В основу разработки положена логически завершенная система взглядов, определяющих оригинальные научно - методические положения по комплексной автоматизации конструкторско - технологического проектирования в условиях мелкосерийного и серийного машиностроения.

Комплексность.

Функциональная структура системы ориентирована на подготовку комплекта технологической документации по вновь осваиваемому изделию или для его технологического сопровождения в производстве.

Универсальность.

Концептуальный, функциональный, программный, технический и организационный комплексы системы не имеют каких - либо принципиальных зависимостей от специфики эксплуатирующих ее предприятий. Собственно проектирование технологических процессов осуществляется с использованием различных форм взаимодействия человека с компьютерной системой. Относительная простота - она отражается в методиках создания и развития программно - информационного комплекса системы, а также быстрой адаптации к системе ее непосредственного пользователя.

Интегрируемость.

Используемой концепцией предусмотрена возможность сквозного и произвольного комплексирования человека - машинных функций системы как по горизонтали (равноподчиненные функции), так и по вертикали (соподчиненные функции) без необходимости в каких либо специальных программно - информационных блоках

сопряжения.

Открытость.

Если раньше автоматизированные процессы проектирования создавались для решения проблем, ориентированных на изделие или учитывающих специфику производства то последние разработки свидетельствуют о все увеличивающейся переориентации на создание универсальных методик и решения задач выходящих за пределы отдельных отраслей производства. Цель такой переориентации - пригодность отдельных систем для различных отраслей. Универсализация базовых систем обуславливает снижение расходов пользователя. Поэтому это свойство САПР обусловлено необходимостью достижения гибкости системы, которое является одним из важных свойств САПР.

Гибкость САПР с точки зрения расширения возможностей ее использования может быть увеличена, если программное обеспечение САПР является универсальным и открытым, в основу структуры таких систем положен принцип - открытой архитектуры, позволяющий адаптировать и развивать многие функции применительно к конкретным задачам и требованиям конечного пользователя.

Каждая САПР имеет свое целевое назначение, т.е. предназначена для решения определенного класса прикладных задач конечного пользователя. Поэтому одной из важных характеристик функционирования САПР является способность системы адаптироваться на конкретные задачи конечного пользователя. Аналогичный вывод сделан американским аналитиком в области САК/САМ Ральфом Гробовским [3], по сведениям которого начиная с 1993 года ведущие производители систем САК/САМ основным требованием к собственным продуктам ставят обеспечение свойства 'используемости' ('Usability'), что означает способность системы адаптироваться на конкретные задачи конечного пользователя. По свойству адаптируемости системы можно разделить на два класса: системы с жесткой архитектурой и системы с открытой архитектурой.

В системах с закрытой архитектурой пользователь не имеет возможности модифицировать программно-математическое обеспечение и адаптация происходит на уровне файлов базы данных посредством заполнения соответствующей информации, т.е. конечный пользователь может адаптировать систему только на уровне информационного обеспечения.

В системах с открытой архитектурой у пользователя есть возможность использования собственных решений не только на уровне

информационного обеспечения, но и на уровне функционирования самой системы посредством изменения программно-математического обеспечения. Такие системы дают более широкие возможности по адаптации на конкретные задачи конечного пользователя, т.к. не требуют вмешательства со стороны разработчика системы.

Степень вмешательства пользователя в виды обеспечения КСАП пропорциональна способности системы адаптироваться на задачи конечного пользователя.

По степени открытости можно выделить следующие поколения систем с открытой архитектурой:

1-е поколение - системы, имеющие настраиваемую конфигурацию. Такие системы строятся по модульному принципу. При этом выделяются отдельно программируемые модули, где пользователь имеет возможность сформировать конфигурацию системы путем выбора отдельных модулей, т.е. осуществить адаптацию программного и технического обеспечения. Основным элементом в таких системах является модуль, который должен удовлетворять требованиям универсальности и информационности. Информационное согласование с другими модулями осуществляется с помощью специальных программ- драйверов, а для информационной согласованности технических модулей используются специальные устройства-адаптеры. Типичными представителями такой системы является САПР электронных схем РСАР.

2-е поколение - системы, в которых адаптация на конкретные задачи конечного пользователя происходит благодаря функционированию системы на основе пользовательских моделей принятия решения. Типичными представителями таких систем являются экспертные системы. Структурно экспертные системы состоят из двух частей:

1. Машина логических выводов.
2. База знаний.

Искусственный интеллект (ИИ) - это программная система, имитирующая на компьютере мышление человека. Для создания такой системы необходимо изучить процесс мышления человека, решающего определенные задачи или принимающего решения в конкретной области, выделить основные шаги этого процесса и разработать программные средства, воспроизводящие их на компьютере. Система ИИ, созданная для решения задач в конкретной проблемной области, называется экспертной системой (ЭС). Источником знаний для наполнения экспертных систем служат эксперты в

соответствующей предметной области. Работа всех экспертных систем основана на экспертной информации, полученной в конкретной предметной области.

Одним из путей улучшения характеристик по производительности и надежности, сократить сроки исследований и разработок новых видов продукции машиностроения, является создание проблемно - ориентированных экспертных систем.

Важным фактором, определяющим возможность использования ЭС, особенно в сфере проектирования, следует считать развитие взаимосвязи между ЭС, и традиционным программным обеспечением. Способность ЭС, взаимодействовать с системами разработки, анализа, базами данных является необходимой для ее успешного функционирования. Целесообразно установление обратной связи ЭС с пользователем для дополнения базы знаний. Возможность расширения базы знаний - главное преимущество ЭС.

При такой архитектуре разработчиками САПР разрабатывается машина логических выводов, а база знаний формируется конечным пользователем, тем самым осуществляется проблемная ориентация системы. Т.о. экспертные системы позволяют адаптироваться на уровне моделей принятия решений.

Развитие ЭС зависит от решения таких проблем , как представление знаний , разработка функциональных языков и машинных способов извлечения знаний, обобщение на естественном языке, построение прототипов ЭС.

Совершенствование ЭС в значительной мере связано с успехами в разработке человеко-машинного интерфейса , технического и программного обеспечения ЭС.

3-е поколение - компилируемые системы- системы с возможностью компиляции исполняемых модулей, что позволяет пользователю адаптировать системы на уровне программно-математического обеспечения, т.е. в таких системах у пользователя есть возможность изменить исходный вариант программно-математического обеспечения.

Как обычно, такие системы делятся на ядро прикладных модулей, которыми решаются основные прикладные задачи и интерфейсные модули, осуществляющие связь прикладных модулей с периферийными устройствами ($hbc=1$). Следовательно, по степени "вмешательства" пользователя в модули системы можно выделить следующие классы систем:

1. Системы с возможностью модификации только на уровне



Исс-1

интерфейсных модулей без ядра прикладных моделей. К таким системам можно отнести системы визуального программирования Kelphi, Visual C, Visual Basic, среда Windows.

2. Системы, в которых присутствовало ядро прикладных задач и вмешательство пользователя осуществлялось только на уровне обеспечения каналов интерфейса. Типичными представителями является Mini Studio, в котором пользователь имеет возможность создавать собственные драйверы для принтера и плотера, осуществить настройку интерактивного режима с пользователем.

3. Системы, позволяющие вмешиваться не только на уровне каналов интерфейса, но и на уровне ядра решения прикладных задач. Типичным представителем является пакет CAKKey.

В статье Ральфа Гробовского "Наступающая революция в CAK", подчеркнута важность двух терминов "Usability"- используемость и "Customizable"- адаптируемость.

Основной смысл, заключенный в этих терминах является следующее: вместо того чтобы покупать полный пакет CAK Key, конечный пользователь может гораздо дешевле выбрать и купить нужные ему приложения в виде отдельных KLL - файлов. Кроме этого посредством добавленных новых функций, конечный пользователь может осуществить настройку интерфейса, полностью или частично изменить внешний вид функционирования системы,

собранный из выбранных пользователем RLL - файлов, имеются также доступы по всем вызовам системы Windows, в которой функционирует САК - пакет. Помимо этого возможно осуществить настройку драйверов плотера, настроить функционирование САК - пакета с базой данных. САК - пакет имеет 3К - систему.

САК Key - это пакет с использованием СкК (САК Key object Keveloper) технологии, обеспечивает ядро графического процессора. Это объектно - ориентированный продукт , созданный на языке C++ и поддерживаемый RXF и RWG форматами и системами Windows и Windows NT.

Корпорация Bentley Systems (Экстон, штат Пенсильвания) анонсировала очередную версию своей системы твердотельного моделирования Micro Station Modeler. В нее включен целый ряд новых функций, а кроме того, в этом продукте впервые реализована возможность “непрерывной модернизации”. Пользователи, подписавшиеся на эту услуг , смогут сообщать о необходимых им усовершенствованиях через Web-страницу Bentley Select. Результаты разработок, выполненных по этим запросам, будут распространяться по Internet, а также на ежеквартально издаваемых компакт-дисках. В качестве отдельного продукта Bentley анонсировала Migration Package, с помощью которого владельцы пакета САКKEY смогут осуществить быстрый и дешевый переход на систему машиностроительного черчения Micro Station. Возможен также быстрый и дешевый переход с САКKEY на ПО твердотельного моделирования Micro Station Modeler. В состав обоих продуктов входит утилита KRAFT-ПАК, предназначенная для преобразования файлов САКKEY в формат Micro Station.

Система Micro Station Modeler способна функционировать на многих платформах, включая ПК, под управлением Windows 3.x, Windows WT, а также машины Apple Power Macintosh, SunSPARC, HPRISC, SGI и REC Alpha.

Важное место среди систем автоматизированного проектирования занимают системы автоматизированного проектирования технологических процессов (САПР ТП). Объектом проектирования САПР ТП является сам технологический процесс, который можно представить в виде системы, которая по степени детализации описания свойств представляет собой иерархически - соподчиненную многоуровневую систему. Следовательно, проектирование такого объекта осуществляется в нескольких этапах и различными подсистемами. Например: САПР маршрутов, САПР отдельных

операций, САПР заготовки, САПР инструментов и т.д.

Технологический процесс в случае ЧПУ (числовое программное управление) детализируется до уровня УП (управляющих программ) для станков с УЧПУ (устройством числового программного управления). Принципиальная особенность станка с ЧПУ заключается в том, что движение инструмента относительно обрабатываемой заготовки задается последовательностью положений и скоростями перемещений его рабочих органов, записываемых в числовой форме на специальном программноносителе.

Внедрение ЧПУ в технологию машиностроения обусловило необходимость построения числовых моделей технологического процесса, а следовательно и широкое использование математических методов и числовой вычислительной техники. Это привело к революционным изменениям в технологии машиностроения. Технология машиностроения из науки, носившей по преимуществу качественный характер, стала превращаться в науку точную. Технологический процесс обработки на станке с ЧПУ в отличие от традиционного технологического процесса требует большей детализации при решении технологических задач и учета специфики представления информации.

Задачи решаемые в САПР УП характеризуются двумя основными свойствами: оригинальность и многовариантность. Следовательно решение задачи в САПР УП должно осуществляться с учетом заранее подготовленных экспертных знаний, которые выражали бы опыт специалистов технологов.

Один из вопросов изучаемых в рамках развития новых методов организации и разработки САПР, представляющих значительный научный и практический интерес - исследование способов представления в САПР ТП технологического опыта независимо от методов, лежащих в основе применяемого принципа проектирования. Любая система действует в соответствии с заложенными в нее алгоритмами проектирования, которые представляют собой элементы знаний специалистов в предметной области и являются формальным выражением их производственного опыта. Поскольку технологическое проектирование зачастую представляет собой набор плохо формализованных эвристик, которые в наибольшей степени подвержены влиянию традиций и нечетко определенной информации, то именно технологические знания, хранящиеся в системе претерпевают склонные изменения в течение ее жизненного цикла, а также при переносе ее с одного производства на другое. Было бы

логично, учитывая специфику технологических знаний и необходимость их постоянного обновления использовать специфику и (как уже отмечалось выше) главное преимущество ЭС - возможность расширять базу знаний.

Однако недостаточный на сегодняшний день уровень формализации технологических задач, незавершенность технологических задач и теоретических основ процессов обработки, неполнота экспериментальных данных не позволяют полностью автоматизировать процессы подготовки технологий и УП для станков с ЧПУ.

Поэтому в большинстве случаев при подготовке УП оптимальной является работа в так называемом диалоговом режиме. В этом режиме наиболее формализованная часть технологических задач решается программно с использованием средств вычислительной техники, а дальнейшие пути решений на узловых участках проектирования выбирает технолог - программист. Это позволяет преодолеть недостаточную формализацию технологических задач, значительно сокращает время проектирования, однако требует творческого участия опытного знающего технолога.

Соответственно, особенностью использования ЭС в машиностроении является необходимость решения задач при наличии неформализованных данных и знаний, а также в условиях лингвистической неопределенности. Кроме того, ЭС должны быть ориентированы на решение таких характерных для машиностроения задач, как диагностика, интерпретация данных, контроль, планирование, проектирование и управление.

Т.о. формализация знаний в ЭС подготовки УП является сложной и трудоемкой задачей, требующей от технолога высокой профессиональной подготовки не только в ряде технических дисциплин, но и в области математики и программной инженерии. Программной инженерией называют инженерно - научную дисциплину, предметом которой является то, что должны знать и уметь лица, занимающиеся различными видами инженерной деятельности по созданию и использованию ПО для профессионально качественного и высокопроизводительного выполнения своей работы.

Формализация вышеуказанных технологических алгоритмов для ЭС подготовки УП требует разработки программных модулей при помощи специальных языков программирования.

Языки описания технологических алгоритмов обладают достаточно ограниченными возможностями т.к. в них наблюдается

отсутствие в большей или меньшей степени алгоритмической полноты. Это означает, что существует ряд классов проектных алгоритмов, нереализуемых в рамках таких языков, т.е. алгоритмов, которые невозможно смоделировать никакими комбинациями средств, имеющихся в данном языке.

Итак, с одной стороны мы имеем технологический процесс, который описывается сложным программно - математическим аппаратом, а с другой стороны язык программирования технологических алгоритмов, имеющий ограниченные алгоритмические возможности.

Поэтому при программировании УП возникают большие трудности, которые в конечном счете оказывают отрицательное влияние на качество производимой продукции, понижается качество и надежность программирования, увеличиваются сроки отладки программных модулей, что увеличивает время на технологическую подготовку производства.

Поэтому архитектура САПР не должна иметь архитектуру экспертной системы, в базе знаний которой экспертные знания формализуются в виде технологических алгоритмов.

Наиболее перспективными в этом отношении представляются идеи абстрактных типов данных (АТД). Основные варианты были предложены во второй половине 70-х годов : (Хар, Лесков, Лондон, Парнас, Шо и др.), на которых основываются идеи объектно-ориентированного программирования (ООП).

Реализовать вышеописанную идею возможно при помощи идеологии ООП созданием системы объектов математических вычислений и отдельно -технологических модулей, при создании которых использовались бы указанные объекты в процессе своего функционирования .

Основные принципы объектно - ориентированной идеологии, это:

1. Инкапсуляция.
2. Наследование.
3. Подклассы.
4. Полиморфизм.

Инкапсуляция - объединение в одном элементе данных и процедур их обработки.

Наследование - сохранение, передача атрибутов данных и выполнение над ними операции от объекта к объекту.

Подклассы - все объекты создаются на основе классов и наследуют свойства и методы классов. Классы могут в свою очередь создаваться

на основе других классов. Такие классы называются подклассами, они наследуют все свойства и методы класса. Подклассы позволяют нам сократить объем программирования и максимально использовать предыдущий опыт работы. Кроме того изменение свойств и методов родительского класса отслеживается и в подклассах. Т.о. можно без особых затрат изменить характеристики всего приложения.

Полиморфизм - в традиционном процедурном программировании аналогом является имя вызывающей подпрограммы.

Исходя из характера задач, решаемых в технологических модулях, и основываясь на объектно - ориентированной идеологии, мы должны определить классы объектов, методы и свойства объектов, характеризующих, описывающих и составляющих технологический процесс. Т.е. выделить составляющие технологического процесса, которые сравнительно легко поддаются формализации и наиболее полно описывают отличительные характеристики указанных составляющих.

При создании программных модулей производится геометрические вычисления для определения $jjghys[njxtr$, составляющих контур детали и эквидистант, вычисления, связанные с определением $jjghys[njxtr$ траектории перемещения инструмента, и вычисления, связанные с расчетом параметров режимов обработки. Следовательно, можно выделить три класса объектов: класс геометрических объектов, класс объектов перемещения и класс объектов расчета технологических режимов обработки. При этом каждый класс должна составлять система объектов с характеризующими данный класс свойствами, и определенными методами использующимися в конкретном классе.

Указанные выше объекты образуют так называемую кВЕСТ - систему. Программные модули, которые программируются технологом и используют объекты кВЕСТ - системы , образуют так называемую X-систему (X-системе). При программировании программных модулей, принадлежащих X-системе , технолог оперирует технологическими значениями, весь сложный программно-математический аппарат присутствует в кВЕСТ - системе , что не требует от технолога специальных знаний в области математики и программной инженерии, достаточно лишь основ программирования.

X-система и кbject-система функционируют в среде ЭС подготовки УП, которая должна включать лишь базовые функции; моделирование траектории перемещения инструмента, визуализации, постпроцессирования и документирования.

Функции объект-системы заключаются в ориентации ЭС на определенные виды механической обработки: токарная, фрезерная, сверлильная, шлифовальная, расточная и т.д. Т.к. указанным видам механической обработки соответствуют различные методы задания контура детали (тел вращения, трехмерные координаты детали, отверстия, корпусные и т.д.), кроме того различают схемы перемещения и геометрию режущего инструмента.

Для вышеописанных видов механической обработки должны существовать собственные математические методы вычисления и как следствие множество объектов, т.е. объект- система.

Описанная архитектура ЭС подготовки УП позволяет создать дополнительные объекты или создать другие объект- системы, и на базе объект- системы дополнительные программные модули в X- системе или создать другие X- системы.

Имея в своем распоряжении систему объектов технолог имеет возможность больше уделить внимания фактической стороне задачи при программировании обработки деталей, т.е. часть задачи которая предполагает создание программно-математического обеспечения значительно облегчается использованием выше описанной системой объектов при создании программных модулей. Т.о. процесс отладки программных модулей значительно упрощается тем что:

- 1) объем листинга программы значительно уменьшается
- 2) текст программы упрощается так как представляет собой простейшие конструкции при программировании.

Китература

1. Гязиров Р.И., Серебряницкий П.П.: Программирование обработки на станках с ЧПУ . Л.Машиностроение 1990г - 591с.
2. Левин Р., Дранг Д., Эндельсон Б.: Практическое введение в технологию искусственного интеллекта и экспертных систем. М."Финансы и статистика" 1991г -235с.
3. Ralpf Grobowski: The Coming Revolution in CAD. CAD++/ Newsletter for developer 1995.

დამუშავების ზონების ავტომატური იდეტიფიკაციის მეთოდის საკითხისათვის

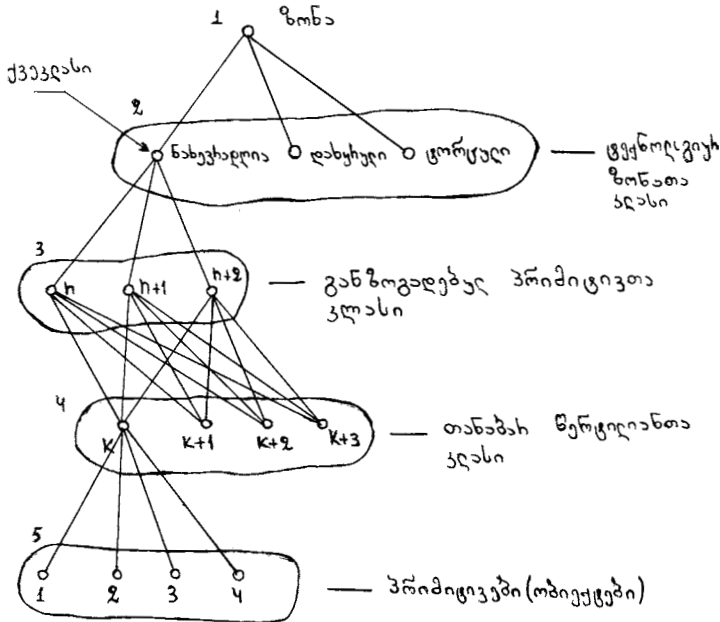
ასპირანტი ა.მამათაყვანძველი

საკონსტრუქტორო ნახაზის სახით მოცემული დეტალის საწყისი აღწერილობის დამუშავება მექანიკური დამუშავების ტექნოლოგიური პროცესების ავტომატიზებული პროექტირებისას მოითხოვს სხვადასხვა ამოცანების გადაწყვეტას:

დეტალის კლასიფიკაცია და მიკუთვნება ამა თუ იმ კომპლექსური დეტალისათვის; საკონსტრუქტორო ნახაზიდან დეტალის ოპერაციული კონტურის გამოყოფა და რედაქტირება; მექანიკური დამუშავების ზონების გამოყოფა და სხვა.

ჩვენს მიერ დამუშავებულ იქნა დეტალის კონტურიდან მექანიკურად დამუშავების ზონების გამოყოფის მეთოდი, რომლის საშუალებითაც შესაძლებელია ამ ეტაპის პროცედურების მნიშვნელოვანი გამარტივება და პროექტირების როგორც პროცესის საიმედობის ამაღლება.

მეთოდის სტრუქტურა საკმაოდ მარტივია და იგი საშუალებას იძლევა ეგმ-ის გამოყენებით დეტალის კონტურიდან გამოიყოს მექანიკური დამუშავების თითქმის ყველა ტიპის ზონა (გარდა ღიასის). ამისათვის,



ნახ. 1. კლასიფიკაცია

ვიყენებთ მექანიკური დამუშავების ზონების კლასებად და ქვეკლასებად დაყოფის მეთოდს, რომელიც წარმოდგენილია გრაფის სახით(ნახ. 1). გრაფის მთავარ ელმენტს წარმოადგენს ზონა(1), რომელიც შედგება სამი ტიპის ქვეკლასისაგან, რომლებიც გაერთიანებულნი არიან ტექნოლოგიურ ზონათა კლასში (2).

ბრუნვითი ტანის დეტალებისათვის შესაძლებელია ტექნოლოგიურ ზონათა ტიპების გამოყოფა იმისდა მიხედვით თუ როგორ იქნება განპირობებული ინსტრუმენტის გადაადგილება. ბრუნვითი ტანის დეტალებისათვის ინსტრუმენტი გადაადგილდება ნახ. 2-ზე მოცემულ კოორდინატთა სისტემაში.

X

-Z

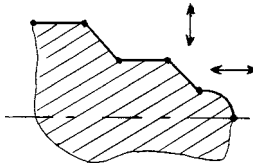
Z

-X

ნახ. 2. სახარატო ჩარხის კოორდინატთა სისტემა

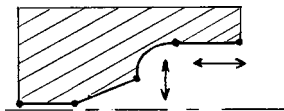
მანქანათმშენებლობის ტექნოლოგიაში განსაზღვრულია [3] ზონათა შემდეგი ტიპები:

1). გარე ნახევრადღია ზონა.



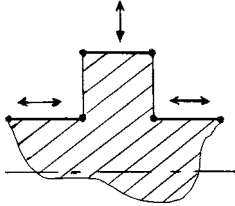
განპირობებს ინსტრუმენტის გადაადგილებას OX და OZ ღერძების მხრიდან.

2) შიდა ნახევრადღია ზონა



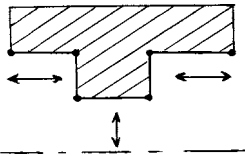
განპირობებს ინსტრუმენტის გადაადგილებას -OX და OZ ღერძების მხრიდან.

3) გარე ღია ზონა



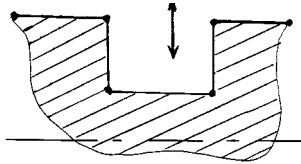
განაპირობებს ინსტრუმენტის გადაადგილებას $-OZ$, OX და OZ ღერძების მხრიდან.

4) შიდა ღია ზონა



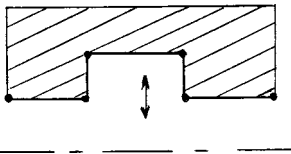
განაპირობებს ინსტრუმენტის გადაადგილებას $-OZ$, $-OX$ და OZ ღერძების მხრიდან.

5) გარე დახურული ზონა



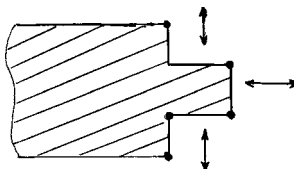
განაპირობებს ინსტრუმენტის გადაადგილებას OX ღერძის მხრიდან.

6) შიდა დახურული ზონა



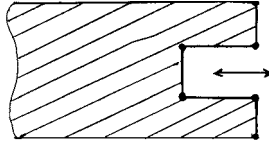
განაპირობებს ინსტრუმენტის გადაადგილებას $-OX$ ღერძის მხრიდან.

7) ღია ტორცული ზონა



განაპირობებს ინსტრუმენტის გადაადგილებას OX, OZ და -OX ღერძების მხრიდან.

8) დახურული ტორცული ზონა



განაპირობებს ინსტრუმენტის გადაადგილებას OZ ღერძის მხრიდან.

ზონების გამოყოფისათვის პრინციპული მნიშვნელობა არა აქვს გარე ნახევრადლია და შიდა ნახევრადლია ზონას, ამიტომ მათ ვაერთიანებთ ნახევრადლიათა ქვეკლასში. ასევე გაერთიანებულია გარე დახურული და შიდა დახურული ზონები დახურულთა ქვეკლასში. ტორცული ზონა გამოყვავით ცალკე ქვეკლასში. ამასთანავე, ღია ზონების გამოყოფა არ სრულდება, რადგან ალგორითმი ფუნქციონირებს მხოლოდ ცილინდრული ნამზადებისათვის.

შედგავდ, ჩვენს მიერ, გამოყოფილ იქნა ტექნოლოგიურ ზონათა კლასი, რომელიც შედგება ნახევრადლია, დახურული და ტორცული ქვეკლასისაგან.

თავის მხრივ, თვითოეული ქვეკლასი(მაგალითად, ნახევრადლია) პრიმიტივების რაოდენობის მიხედვით იყოფა ქვეკლასებად, რომლებიც გაერთიანებულნი არიან განზოგადებულ პრიმიტივთა კლასში (3). ეს იმას ნიშნავს, რომ განზოგადებულ პრიმიტივთა კლასის ყოველ ქვეკლასში მოთავსებულია ნახევრადლია ქვეკლასის ერთი პრიმიტივი და ქვეკლასების რაოდენობა ტოლია ნახევრადლია ქვეკლასში პრიმიტივების რაოდენობისა.

განზოგადებულ პრიმიტივთა კლასიდან გამოიყოფა თანაბარწერტილიანთა კლასი (4), რომელიც შედგება ქვეკლასებისაგან. ამასთან თითოეულ ქვეკლასში მოთავსებულია განზოგადებულ პრიმიტივთა კლასის ქვეკლასებიდან გამოყოფილი თანაბარი სიდიდის (საყრდენი წერტილების რაოდენობის მიხედვით) პრიმიტივები. და ბოლოს, ვიღებთ თანაბარწერტილიანთა კლასის ქვეკლასის პრიმიტივებს (5).

განვიხილოთ ზონების გამოყოფის ალგორითმი, რომელიც ფუნქციონირებს ზემოთ განხილულ კლასიფიკაციაზე დაყრდნობით. ამასთან აქ ვიყენებთ მიდგომას, რომელიც მდგომარეობს გარკვეული პირობების გათვალისწინებით, ასე ვთქვათ, დიდი ბიჯის გადადგმაში და შესაძლო მაქსიმალური ზონის გამოყოფაში. როგორც მუშაობის მსვლელობამ გვიჩვენა, თუ გამოვყოფთ შესაძლო მაქსიმალურ ზონას (გადავდგამთ

დიდ ბიჯს) და მისით დავიწყებთ პრიმიტივებთან შედარების პროცესს, არ შედარების შემთხვევაში მისი თანდათან შემცირების ხარჯზე, თავიდან ავიცილებთ მრავალ არასასურველ იტერაციას, დროის მცირე მონაკვეთში ნაკლები შრომის ფასად გამოვყოფთ მაქსიმალურ ზონას და ამასთან ეს ზონები ტექნოლოგიური თვალსაზრისით მისაღებია.

ზონების გამოყოფის მექანიზმი ნაჩვენებია ნახ. 4-ზე. მეთოდის ძირითადი თავისებურება მდგომარეობს იმაში, რომ ზონების გამოყოფა ხორციელდება წინასწარ აღწერილი ტიპური ზონებით. ამისათვის თავდაპირველად ხდება დეტალის კონტურზე სავარაუდო კონტურის გამოყოფა, ანუ გარკვეული ჰიპოთეზის დაშვება, შემდეგ ხდება ჰიპოტეზის შემოწმება ანუ სავარაუდო კონტურის შედარება ყველა შესაძლო ტიპურ ზონებთან. თუ შედარება დადებითია მაშინ ხდება სავარაუდო კონტურიდან ზონის გამოყოფა. წინააღმდეგ შემთხვევაში ხორციელდება ახალი სავარაუდო კონტურის გამოყოფა ანუ ახალი ჰიპოთეზის დაშვება.

თიხიური
ზონები

ჰიპოთეზა

შედარება

აღწერა

ნახ. 4. ზონების გამოყოფის მექანიზმი

მათემატიკური თვალსაზრისით, ეს ამოცანა დაიყვანება გადარჩევის ამოცანად, როდესაც წინასწარ განსაზღვრულია შესაძლო ტიპური გადაწყვეტები და სინთეზის ამოცანა წარმოადგენს გარკვეული მოსაზრებით (კრიტერიუმით) საუკეთესო ალტერნატიული ვარიანტის ამორჩევას. მათემატიკურად აღნიშნული ამოცანა ჩაიწერება შემდეგი კორტეჟის სახით:

$$\alpha = \{\Omega, N\}$$

სადაც α მიღებული გადაწყვეტილება, Ω ამორჩევის კრიტერიუმი, ხოლო N შესაძლო ალტერნატიული ვარიანტების სიმრავლეა.

როგორც ცნობილია გადარჩევის ამოცანა გამოირჩევა შედარებითი სიმარტივით, მაგრამ აქვე უნდა აღინიშნოს რომ შესაძლო ალტერნატიული ვარიანტების დიდი სიმრავლის გამო მეთოდის რეალიზება ხშირად ძალზედ პრობლემატურია. ამიტომ უმრავლეს შემთხვევაში საქმე გვაქვს მიზანმიმართულ გადარჩევამდე, როდესაც ამორჩევის კრიტერიუმის გარდა

ხორციელდება ე.წ. ძებნის სტრატეგიის განსაზღვრა. ამ შემთხვევაში განსაკუთრებული მნიშვნელობა ენიჭება სწორი სტრატეგიის განსაზღვრას, ვინაიდან წინააღმდეგ შემთხვევაში, მიღებული კრიტიკერიუმის თვალსაზრისით, შესაძლოა არასასურველი გადაწყვეტილების შერჩევა.

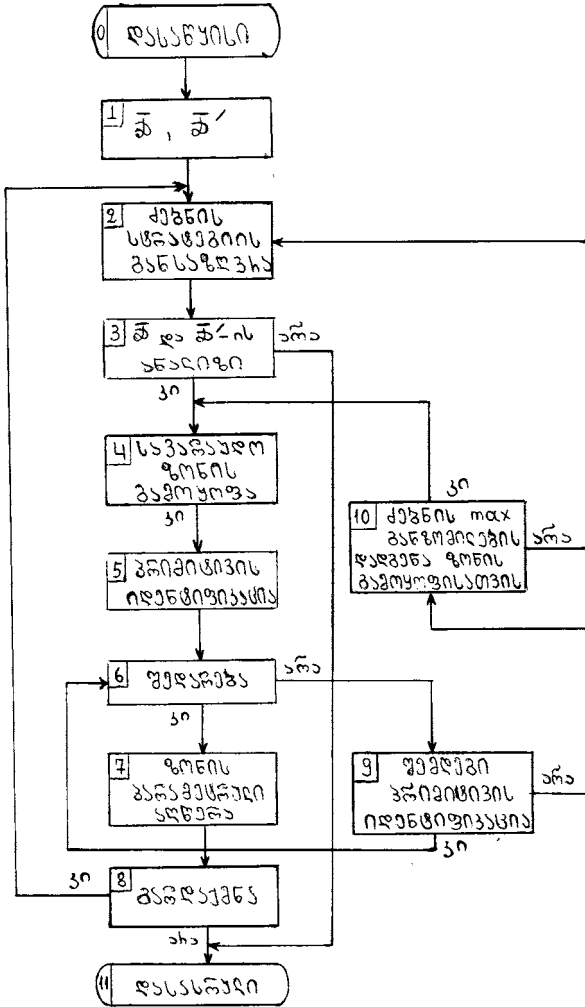
ჩვენს შემთხვევაში საკონსტრუქტორო პრიმიტივების შესაძლო დიდი სიმრავლის გამო მიზანშეწონილია მიზანმიმართული გადარჩევის განხორციელება, რაც შეგვიძლია წარმოვიდგინოთ როგორც სახეთა გამოცნობის ამოცანა [1], რომელიც დაყრდნობილია თვისებათა საერთობის პრინციპზე [4] და რომელიც სავარაუდო ფრაგმენტის თვისებების გათვალისწინებით საკონსტრუქტორო პრიმიტივების (ეტალონთა) საერთო მასიდან, მათი თვისებების შედარების საფუძველზე, გამოყოფს ოპტიმალურს და აღნიშნულ ფრაგმენტს მიაკუთვნებს მას.

ზონების გამოყოფის პროცესში ხდება დეტალის და ნაშადის კონტურის ეტაპობრივი გარდაქმნა, ამიტომ კონტურების შუალედური მდგომარეობის დასახასიათებლად შემოღებულია ე. წ. მდგომარეობის ვექტორი, რომელიც აღწერს დეტალის და ნაშადის მიმდინარე კონტურებს (ნახ.5, ბლოკი 1), ამასთან ალგორითმი ფუნქციონირებს მხოლოდ ცილინდრული ტიპის ნაშადებისათვის, რაც წარმოადგენს შეზღუდვას მეთოდისათვის. თუმცა, უნდა აღინიშნოს რომ სხვა ტიპის ნაშადები(ნაშტამპი და ჩამოსხმები) ძირითადად გამოიყენება მასიური ტიპის წარმოებაში თავისი სიძვირის გამო, ხოლო ცილინდრული სახის ნაშადები დამზადების შედარებით მარტივი ტექნოლოგიის ხარჯზე უფრო იაფია და შედეგად უფრო ხშირად გამოიყენება წვრილსერიულ და საშუალო სერიულ წარმოებაში.

თავდაპირველად ხდება ძებნის სტრატეგიის განსაზღვრა. ვადგენთ გამოსაყოფი ზონის ტიპს(ნახევრადლია, ტორცული, დახურული) ტექნოლოგიურ ზონათა კლასიდან (ნახ. 1) და ვპოულობთ მაქსიმალური საყრდენი წერტილის სიდიდეს(ბლოკი 2). როგორც ვაჩვენა ჩატარებულმა ანალიზმა, ტექნოლოგიურ ზონათა კლასში ქვეკლასების ძებნის სტრატეგიის განსაზღვრა უნდა იყოს შემდეგნაირი: ჯერ უნდა გამოიყოს ნახევრადლია ზონები, შემდეგ დახურული და ბოლოს ტორცული. ნებისმიერი სხვა მიმდევრობით აღნიშნული ქვეკლასების გამოყოფას მიყვევართ ტექნოლოგიური თვალსაზრისით არარეალური ზონების გამოყოფამდე.

ამის შემდეგ ხორციელდება მდგომარეობის ვექტორის ანალიზი (ბლოკი 3). იმ შემთხვევაში თუ დეტალის კონტურზე არის კიდევ აღსაწერი კონტურები, მაშინ გადავდივართ შემდეგ ბლოკზე, თუ არა დავასრულებთ მუშაობას (ბლოკი 11).

მდგომარეობის ვექტორის ანალიზის შემდეგ, ტექნოლოგიურ ზონათა ქვეკლასისა და ამ ქვეკლასისათვის ძებნის მაქსიმალური განზომილების მიხედვით გამოვყოფთ სავარაუდო ზონას და ვადგენთ



ნახ. 5. ფუნქციონალური ბლოკ-სქემა

ძეხნის განზომილებას პრიმიტივის იდენტიფიკაციისათვის (ბლოკი 4). თუ სავარაუდო ზონის გამოყოფა ვერ მოხერხდა თავიდან ვიწყებთ ძეხნის ახალი სტრატეგიის განსაზღვრას ანუ ტექნოლოგიურ ზონათა კლასში გადავიდებით შემდეგ ქვეკლასზე (ბლოკი 2). ვახდენთ პრიმიტივის იდენტიფიკაციას (ბლოკი 5), ანუ ვაგებთ ნახ. 2-ზე მოცემულის მსგავს

გრაფს კონკრეტული ამოცანისათვის შემდეგი თანმიმდევრობით, ჯერ ნახევრადლია, შემდეგ დახურული, ბოლოს ტორცული და ვირჩევთ პრიმიტივებს (ობიექტებს) პრიმიტივის იდენტიფიკაციისათვის ძენის განზომილების მიხედვით.

ამრიგად, ამ ეტაპისათვის სავარაუდო ზონა გამოყოფილია და შედარებისათვის საჭირო პრიმიტივებიც ნაპოვია. ამ შემთხვევაში აუცილებელია მოხდეს გამოყოფილი ზონის შედარება პრიმიტივებთან, ანუ რაც იგივეა მისი ერთერთისათვის მიკუთვნება (ბლოკი 6). იმ შემთხვევაში თუ პირველ პრიმიტივთან შედარება ვერ განხორციელდა გადავდივართ შემდეგი პრიმიტივის იდენტიფიკაციაზე იგივე ძენის განზომილებისათვის (ბლოკი 9) და თავიდან ვახდენთ შედარებას (ბლოკი 6). თუ მოხდა ისე, რომ არცერთი პრიმიტივი არ შეუდარდა ძენის განზომილებას ვამცირებთ ერთით (ბლოკი 10) და თავიდან ვიწყებთ სავარაუდო ზონის გამოყოფას (ბლოკი 4), ხოლო თუ ძენის განზომილების შემცირება შეუძლებელია მაშინ თავიდან ვიწყებთ ძენის ახალი სტრატეგიის განსაზღვრას (ბლოკი 1), ანუ ტექნოლოგიურ ზონათა კლასში გადავდივართ შემდეგ ქვეკლასზე. თუ შედარება წარმატებით განხორციელდა (ბლოკი 6) ვახდენთ გამოყოფილი ზონის პარამეტრიზებულ აღწერას, ანუ ვითვლით ზონის პარამეტრების რიცხვით მნიშვნელობებს (ბლოკი 7).

ამის შემდეგ, ვახდენთ მდგომარეობის ვექტორის გარდაქმნას (ბლოკი 8). დეტალისა და ნამზადის კონტურიდან ვიღებთ აღწერილ მონაკვეთებს და დარჩენილი კონტურისათვის ვანხორციელებთ აღწერას თავიდან (ბლოკი 2) და მთელი ციკლი მეორდება მანამ, სანამ დეტალის მთლიანი კონტური არ აღიწერება, ან მანამ, სანამ დეტალის კონტურის აღწერა შეუძლებელი გახდება. ამის შემდეგ სისტემა დაასრულებს მუშაობას (ბლოკი 11).

ლიტერატურა

1. ა.შარმაზანაშვილი, ა. მამათავრიშვილი. დეტალის გეომეტრიის გარდაქმნა სახეთა გამოცნობის მეთოდით. \სტუ-შრომები. 1(417). თბილისი — გვ. 161—169. N 0201-7164. S S
2. Разработка УП с помощью микро ВМ s i- t e k
1987.-3-N . 0 6 1 — 8 4 1
н З и б я р е . Д А н , о . з Д р я е и т н .
е Д , х е а
1989. — 288с.
4. Дж .Ту, Р. Гонсалес. Принципы распознавания образов. 1978

Разработка модуля инвариантного постпроцессирования управляющих программ в САПР ТП

аспирант Н Долидзе

Эффективность компьютеризированных производственных систем (КПС) в большой степени зависит от уровня и качества решения проблемы автоматизации программирования систем числового программного управления (СЧПУ) для различных видов технологического оборудования - токарных, фрезерных и других станков.

Числовое программное управление (ЧПУ) это процесс во время которого происходит управление технологическим процессом производства посредством управления специальными числовыми кодами, последовательность которых образуют так называемую управляющую программу (УП). Команды УП определяют траекторию движения и режимы работы исполнительных органов технологического оборудования.

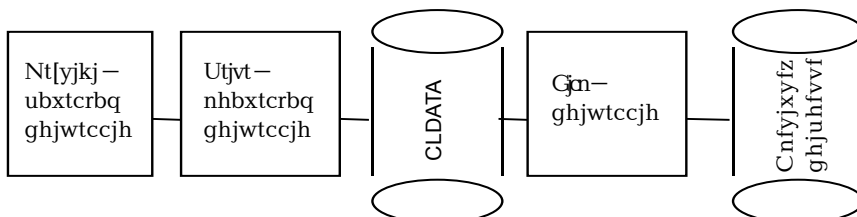
Создание УП осуществляется на этапе технологической подготовки производства при помощи специальных систем автоматизированного проектирования (САПР УП).

В состав САПР УП входит три основных модуля:

1. Технологический процессор
2. Геометрический процессор
3. Постпроцессор.

Взаимосвязь описанных модулей представлена на рисунке.

Технологический процессор предназначен для определения параметров режима работы исполнительных органов технологического оборудования, таких как например: скорость перемещения, количество оборотов шпинделя и т. д.



CLKATA (Cutter Location Kata) является промежуточным уровнем описания УП, который представляет собой массив содержащий последовательность логических записей переменной длины. Каждая запись это последовательность логических слов содержащая информацию определенного вида, например: координаты, определяющие траекторию движения инструмента, указатели технологических режимов обработки, канонические формы геометрических элементов контура, сведения о допусках, о режущем инструменте и т. д.

Станочная управляющая программа представляет собой второй уровень описания УП для конкретного станка с ЧПУ.

Постпроцессор является программным модулем предназначенным для обработки записей CLKATA и выдающем на выходе станочную программу для каждого отдельного устройства ЧПУ, кроме того в функции постпроцессора входит осуществление т. н. обратной компиляции из станочной программы в CLKATA, что позволяет верифицировать визуальными средствами САПР УП, уже инеющуюся станочную программу.

Хотя международным стандартом ISk и регламентированны определенные правила кодирования числовой информации в станочных УП, тем не менее ISk стандарты не всегда соблюдаются.

Станочную программу записывают в виде последовательности кадров (содержание кадров станочной программы также регламентировано). Информационные слова в кадре обычно записывают в следующей последовательности:

1. Подготовительная функция - (G);
2. Размерные перемещения по различным координатам - (X,Y,Z,U,V,W,P,Q,A,B,C);
3. Параметры интерполяции или шаг резьбы - (I,J,K);
4. Функция подачи - (F);
5. Функция главного движения - (S);
6. Функция инструмента - (T);
7. Вспомогательная функция - (M);

Кроме того , в разных устройствах ЧПУ некоторые команды имеют одинаковое функциональное назначение. Как правило, это команды подготовительной (группа G) и вспомогательной (группа M) функций, например:

G00 быстрый ход

G01 линейная интерполяция

G02 круговая интерполяция по часовой стрелке

G03 круговая интерполяция против часовой стрелке
G90 ввод абсолютного значения
G91 ввод относительного значения
G94 подача на мм/мин
G95 подача на мм/об
M03 вращение шпинделя по часовой стрелке
M04 вращение шпинделя против часовой стрелке
M05 останов шпинделя.

Остальные команды разных устройств ЧПУ могут отличаться не только форматом записи, но и назначением. Кроме того команды обозначающие определенное действие у одних УЧПУ могут совершенно отсутствовать у других устройств ЧПУ.

Внизу приведены примеры демонстрирующие различия командных слов в следующих устройствах ЧПУ:

- Sinumerik 7T
- CNC - Y
- Huror PNC 712
- Электроника НЦ 31-01.

F - функция подачи

Sinumerik 7T	CNC-Y	Huror PNC 712	YW31
<i>F023</i>	<i>F05</i>	<i>F1.3</i>	<i>F06</i>

Данная таблица демонстрирует разность форматов записи функции подачи у разных УЧПУ, например: у Sinumerik 7T формат *F023*, *lfyujt* означает что во время записи функции подачи после *F* командного слова пишется численное значение параметра, где целое число не должно превышать 2 единицы, а дробная часть - 3 единицы, кроме того 0 обозначает что в случае если целое число это целое двузначное число то на месте сотенного числа пишется 0 (напр: *F01.706*).

M02 - конец программы

Как видно из таблицы у Sinumerik 7T, CNC - Y и Электроника НЦ 31-01 функция *M02* определяет конец программы, но эта же самая функция отсутствует у Huror PNC 712.

Подготовительная функция G60

В таблице демонстрируется разное функциональное назначение подготовительной функции G60:

- Sinumerik 7T - подготовительная функция G60 определяет точный останов
- CNC - Y и Электроника НЦ 31-01 данная функция отсутствует
- Hupor PNC 712 - подготовительная функция G60 имеет другое функциональное назначение и определяет организацию цикла.

M06 - смена инструмента

Функциональное назначение M06 функции это смена инструмента.

Sinumerik 7T	CNC-Y	Hupor PNC 712	YW31
<i>ij nce nndet n</i>	<i>ij nce nndet n</i>	<i>j nce nndet n</i>	<i>j nce nndet n</i>

и Электроника
12 и Электроника

M46 - работа с инструментальным магазином

Как видно из таблицы M46 вспомогательная функция отсутствует у Sinumerik 7T, Hupor PNC 712 и Электроника 31-01.

Так как, основной задачей постпроцессора является создание станочной программы, то различие команд в разных устройствах ЧПУ порождает проблему постпроцессоров для разных САПР УП, состоящую в том, что каждое новое технологическое оборудование требует разработки соответствующего постпроцессорного модуля.

Решение данной проблемы возможно двумя способами:

- Создание новых постпроцессорных модулей разработчиками САПР УП. Такое решение характерно для САПР УП с жесткой архитектурой, при этом адаптация программно-математического обеспечения (ПМО) на конкретный парк

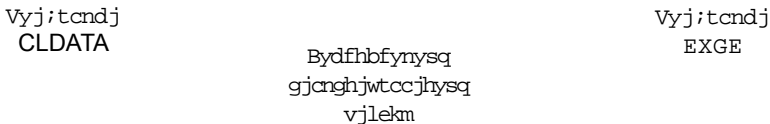
- УЧПУ пользователя не может быть осуществлена самим пользователем.

Настройка постпроцессорного модуля на конкретное технологическое оборудование самим пользователем, что соответствует концепции САПР УП с открытой архитектурой.

Создание САПР УП с открытой архитектурой приводит к необходимости разработки инвариантного постпроцессора, настраиваемого на конкретную модель УЧПУ. Однако применение такого инвариантного постпроцессора снимает проблему в рамках определенных САПР УП. Как обычно САПР УП отличаются правилами описания промежуточного языка CLKATA. Несмотря на определенные унификации промежуточные данные CLKATA различных САПР УП (например: АРТ, МРКАРТ, ЕХАРТ, 2СL, ТЕХТРАН, АРТ ЕС, АРТ СМ, САПС - 9 И ДР.) имеют свои особенности: количество слов в записи, порядок описания, разность форматов, разность кодов; в результате возникает необходимость разработки собственного инвариантного постпроцессора для конкретной САПР УП.

Исходя из вышесказанного, возникла идея создания такого инвариантного постпроцессора, который представлял бы из себя независимый модуль адаптируемый к различным САПР УП в базе данных которой будут заложены модифицируемые данные о различных устройствах ЧПУ и о CLKATA различных САПР УП.

Как видно из следующего рисунка модуль инвариантного постпроцессора выделен из САПР УП и служит связующим звеном между САПР УП и группой УЧПУ.



Процесс разработки инвариантного постпроцессора можно разделить на следующие основные этапы:

1. Разработка методики компиляции станочной программы.
 2. Разработка методики компиляции CLKATA из станочной программы.
 3. Разработка базы данных инвариантного постпроцессора.
- Рассмотрим вышесказанные этапы подробно.

Методика компиляции станочной программы может быть представлена в виде следующих последовательных шагов, которые позволяют считывать и идентифицировать запись CLKATA и в итоге формировать станочную программу:

1. Считывание записи CLKATA
2. Идентификация формата CLKATA
3. Определение функционального назначения записи CLKATA
4. Идентификация параметров и вспомогательных слов CLKATA
5. Идентификация продолжения записи CLKATA
6. Нахождение соответственной станочной функции
7. Определение формата записи слов станочной программы
8. Определение последовательности слов в станочной программе
9. Формирование параметров адресных слов станочной программы
10. Запись кадра станочной программы
11. Идентификация наличия следующей записи CLKATA .
12. True: переход на пункт 1.

Методика компиляции CLKATA из станочной программы состоит из следующих шагов:

1. Считывание командного слова в кадре станочной программы.
2. Идентификация формата командных слов станочной программы.
3. Определение функционального назначения командного слова станочной программы.
4. Нахождение соответственного функционального назначения записи CLKATA.
5. Определение формата записи CLKATA.
6. Идентификация параметров и вспомогательных слов CLKATA.
7. Формирование записи CLKATA.
8. Идентификация формирования многострочной записи
9. True: переход на пункт 1.

Разработку базы данных инвариантного постпроцессора упрощенно

можно разделить на два основных этапа:

- Разработка концептуальной модели базы данных.
- Разработка и создание физической модели базы данных.

Во время разработки концептуальной модели базы данных было рассмотрено и апробировано несколько подходов.

На ранней стадии разработки концептуальной модели возникла идея создания информационного массива для разных групп командных слов всех моделей УЧПУ, например: информационный массив “Подготовительная функция” должен иметь следующие поля:
{Функция} {код УЧПУ} {код CLKATA} {под-код CLKATA}

Где:

- Функция - Означает командное слово
- Код УЧПУ - Код присвоенный конкретному станку
- Код CLKATA - Главный код CLKATA
- Под-код CLKATA - Код вспомогательного слова CLKATA.

Кроме того существует еще один информационный массив связывающий имя УЧПУ с кодом станка, со следующими полями:

{Станок} {Код станка}

Где :

- Станок - Означает имя УЧПУ
- Код станка - имя конкретного УЧПУ употребляемый в массивах командных слов

При данном подходе, для того чтобы в результате компиляции получить станочную программу в постпроцессоре следует жестко определить последовательность чтения информационных массивов, например:

1. Номер кадра;
2. Массив подготовительной функции;
3. Массив размерных перемещений;

и т. д.

Кроме того, количество информационных массивов небольшое, но сами массивы довольно большого объема так как они содержат информацию о всех моделях УЧПУ.

Данный метод также противоречит разработанной методике компиляции станочной программы из CLKATA, например:

1. Не учитывается модальность командных слов (модальное командное

слово действует и на последующий кадр), так как отсутствуют соответственные информационные поля, которые учитывают модальность функций.

2. Не возможна идентификация формата CLKATA (отсутствует информационное поле учитывающее формат CLKATA).

3. Отсутствует идентификация параметров и вспомогательных слов (отсутствуют информационные поля описывающие параметры и вспомогательные слова).

4. Отсутствует идентификация продолжения записи (отсутствуют соответственные информационные поля).

Другая отрицательная сторона разработанной концептуальной модели это отсутствие возможности компиляции CLKATA из станочной УП, т. е. Не учтена разработанная методика компиляции, так как отсутствуют информационные поля полностью описывающие CLKATA.

На втором этапе разработки в отличие от предыдущей концептуальной модели было решено создать отдельный информационный массив для каждого отдельного УЧПУ, содержащий описания каждого информационного слова данного оборудования. Такой информационный массив должен иметь следующие поля:

{Код информационного слова} {Псевдо-код} {Информационное слово}

Где:

- Код информационного слова - определяет принадлежность последующих командных слов к определенной группе;
yfgbhvth: G - принадлежность к подготовительным функциям
- Псевдо-код - код назначенный постпроцессором для связывания кода CLKATA и определенного командного слова УЧПУ;
- Информационное слово - командное слово УЧПУ, напр. G00

Кроме того существует информационный массив связывающий псевдо-коды с кодами CLKATA, со следующими полями:

{Псевдо-код} {код CLKATA}

Где :

- Псевдо-код - код назначенный постпроцессором для связывания кода CLKATA и определенного командного слова УЧПУ;
- код CLKATA - код главного слова CLKATA.

Третий информационный массив связывает псевдо-коды со словами АПТ и имеет следующие поля:

$\{Псевдо-код\} \{слово АПТ\}$

Где:

- Псевдо-код - код назначенный постпроцессором для связывания кода CLKATA и определенного командного слова УЧПУ;
- Слово АПТ - информационное слово определяющее функциональное назначение командного слова или записи CLKATA;

При данном подходе опять не учитывается модальность функций и разработанные меодики компиляции CLKATA из станочной УП и станочной УП из CLKATA, так как отсутствуют соответственные информационные поля а также увеличивается количество информационных массивов.

На третьем этапе разработки концептуальной модели базы данных в информационный массив УЧПУ было добавленно несколько информационных полей. В результате информационный массив имеет следующие поля:

$\{Код информационного слова\}$

$\{Псевдо-код\} \{Модальность\} \{Формат\} \{Информаци. слово с форматом\}$

Где:

- Код информационного слова - определяет принадлежность последующих командных слов к определенной группе
- Псевдо-код - код назначенный постпроцессором для связывания кода CLKATA и определенного командного слова УЧПУ
- Модальность - определяет модальность данного информационного слова, напр. 0-не модальный, 1-модальный
- Формат - определяет существование параметров у данного командного слова
- Информационное слово с форматом - определяет само командное слово и его формат если таковой существует;

Информационный массив для CLKATA имеет следующие поля:

$\{Код CLKATA\} \{Под код CLKATA\} \{Псевдо-код\}$

Где:

- Код CLKATA - код главного слова CLKATA
- Под код CLKATA - код вспомогательного слова CLKATA
- Псевдо-код - код назначенный постпроцессором для

связывания кода CLKATA и определенного командного слова УЧПУ.

Данный метод не учитывает разработанную методику компиляции CLKATA из станочной УП, слова УЧПУ.

Данный метод не учитывает разработанную методику компиляции CLKATA из станочной УП, так как:

1. Не определяется формат записи CLKATA
2. Не идентифицируются параметры слов CLKATA.

В итоге невозможно сформировать запись CLKATA, поэтому возникла потребность сохранить дополнительную информацию о CLKATA и как результат модифицировать массив CLKATA.

Исходя из сказанного на следующем этапе разработки поля массивов каждого УЧПУ остаются без изменений, но добавляется новый информационный массив связывающий псевдо-код со словами АПТ. Этот массив имеет следующие поля:

{Псевдо-код} {слово АПТ}

Где:

- Псевдо-код - код назначенный постпроцессором для связывания кода CLKATA и определенного командного слова УЧПУ
- Слово АПТ - Командное слово АПТ.

Как сказано выше информационный массив CLKATA модифицируется и имеет следующие поля:

Псевдо-код = {(код инфор. слова, место А, формат А, номер строки, код CLKATA, место В, формат В, параметр 1, место 1, формат 1, параметр 2, место 2, формат 2, . . . , параметр n, место n, формат n)}

Где:

- Псевдо-код - код назначенный постпроцессором для связывания кода CLKATA и определенного командного слова УЧПУ
- Код инфор. слова - код информационного слова CLKATA
- Место А - местоположение информационного слова CLKATA
- Формат А - формат информационного слова CLKATA
- Номер строки - номер записи CLKATA
- Код CLKATA - Код командного слова того слова CLKATA
- Место В - местоположение командного слова CLKATA
- Формат В - формат командного слова CLKATA
- Параметр n - параметр командного слова CLKATA

- Место *n* - местоположение параметра командного слова CLKATA
- Формат *n* - формат параметра командного слова CLKATA.

Данный метод имеет существенный недостаток - дублирование записей в информационном массиве CLKATA, поэтому возникла необходимость изменения существующего информационного массива CLKATA на более совершенный массив. В итоге реконструированный информационный массив CLKATA содержит следующие поля:

Псевдо-код = {*Главная часть*} [*Описание*] { *Параметры* };

Где:

- Псевдо-код - код назначенный постпроцессором для связывания кода CLKATA и определенного командного слова УЧПУ
- Главная часть - описывает код информационного слова CLKATA, местоположение информационного слова CLKATA, формат информационного слова CLKATA, код командного слова CLKATA, местоположение командного слова CLKATA, формат командного слова CLKATA;
- Описание - анализирует вспомогательные слова главного командного слова CLKATA;
- Параметры - описывает параметры командного слова CLKATA, местоположение параметра командного слова CLKATA, формат параметра командного слова CLKATA, номер записи CLKATA;

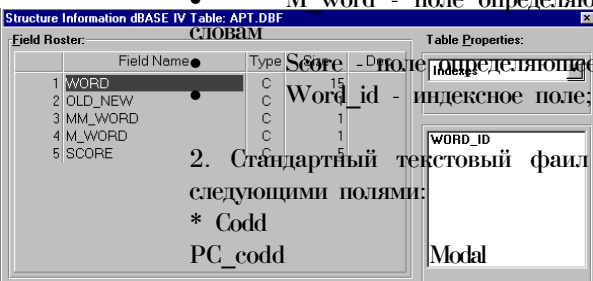
Следующим этапом разработки базы данных является разработка физической модели базы данных инвариантного постпроцессора. Физическая модель базы данных инвариантного постпроцессора включает следующие файлы:

1. Файл информационных слов АИТ разработанный в стандартной среде dBaseIV с расширением dbf. Структура данного файла показана на следующем рисунке:

Где:

- Field name - имя поля

- Type - тип поля
- Size - размер поля
- C - обозначает символьный тип;
- Word - информационное слово АПТ;
- kld_new - поле определяющее принадлежность к немодифицируемым словам;
- MM_word - поле определяющее принадлежность к информационным словам
- M word - поле определяющее принадлежность к главным



Score - поле определяющее псевдо-код
 Word_id - индексное поле;
 2. Стандартный текстовый файл УЧПУ с расширением txt со следующими полями:
 * Codd
 PC_codd
 Modal
 Format Fword

Где :

- * - символ отделяющий различные группы командных слов
- Codd - определяет принадлежность последующих командных слов к определенной группе
- PC_codd - код назначенный постпроцессором для связывания кода CLKATA и определенного командного слова УЧПУ;
- Modal - определяет модальность данного информационного слова, напр. 0-не модальный1-модальный.
- Format - определяет существование параметров у данного

командного слова

- Fword - определяет само командное слово и его формат если таковой существует;

3. Стандартный текстовый файл CLKATA с расширением txt со следующими полями;

Code={Main} [Ques] {Kesc}

Где:

- Code - Псевдо-код
- Main - описывает код информационного слова CLKATA, местоположение информационного слова CLKATA, формат информационного слова CLKATA, код командного слова CLKATA, местоположение командного слова CLKATA, формат командного слова CLKATA ;
- Ques - анализирует вспомогательные слова главного командного слова CLKATA;
- Kesc - описывает параметры командного слова CLKATA, местоположение параметра командного слова CLKATA, формат параметра командного слова CLKATA, номер записи CLKATA;

В свою очередь поля Main, Qeus и Kesc содержат собственные под-поля.

Main содержит:

- P1=P1 T:Cn:F;
- P2=P2 T:Cn:F;

Где:

P1 - основное слово CLKATA,
P2 - главное слово CLKATA,
T - тип слова,
Cn - местоположение слова
F - формат слова.

Qeus содержит следующую строку:

- Cn=sbcode1 code, Cn= sbcode2 code+1;

Где:

Cn - местоположение слова
sbcode1 - код вспомогательного слова

sbcode2 - код вспомогательного слова;
code - псевдо-код
code+1 - измененный псевдо-код;

Kesc содержит следующую строку:

- K1/K2/.../Kn;

Где:

Kn в свою очередь содержит информацию о параметрах и номере строки

$K_n = N(n) - \{Pole\}$;

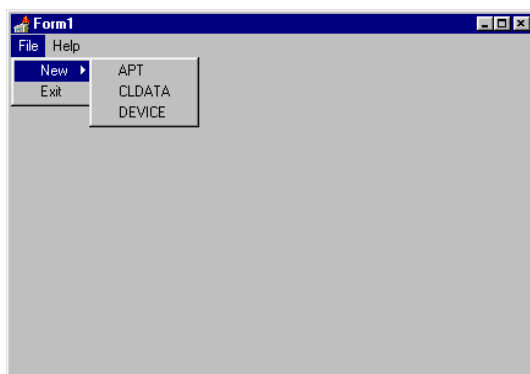
$N(n)$ - определяет номер строки;

Pole - содержит информацию о параметрах;

$Pole = \{a():Cn:F / x():Cn:F / y():Cn:F / z():Cn:F /$
 $ztool():Cn:F / I():Cn:F / J():Cn:F / K():Cn:F /$
 $R():Cn:F / nst():Cn:F / nm():Cn:F\}$

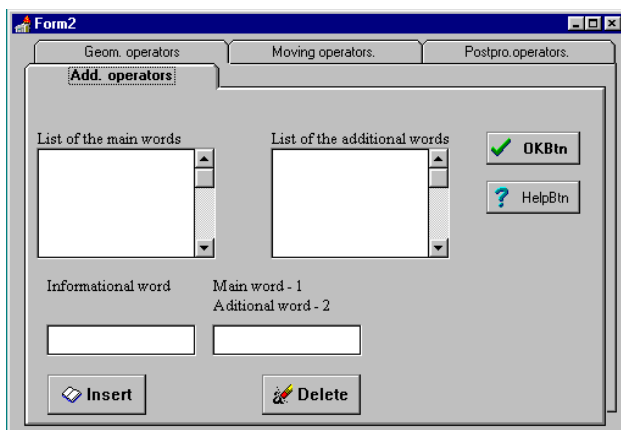
- a - любой параметр
- x,y,z - x,y,z координаты;
- I,J,K - I,J,K координаты;
- R - координаты;
- nn - номер строки CLKATA;;
- nst - количество слов в записи CLKATA;

Для заполнения базы данных была создана интерактивная система в визуальной среде разработки прикладных программ Windows - Kelphi. Главное меню в интерактивном режиме имеет следующий вид:



Выбирая опцию “New” пользователь получает возможность редактировать базу данных слов АПТ, CLKATA и УЧПУ соответственно выбирая опции АРТ,CLKATA и KEVICE.

Если выбрана опция АПТ, то форма главного меню закрывается и появляется форма позволяющая модифицировать существующий набор слов АПТ, например добавлять, изменять и стирать слова. Данная форма показанна на следующем рисунке:



Ds,bhfv pfrkflre jghtltkztnz ghbyflkt;yjcnm ckjdf

r rjyrhtnyjq uheggt jgthfnjhjd (yfgH= ghbyflkt;yjcnm
r uheggt utjvtnhbxtcrbv jgthfnjhfv)=

List of the main words - выходит список (перечень) главных слов;

List of additional words - выходит список (перечень) вспомогательных слов

Informational word - ввод желаемого слова;

Main word - 1; additional word - 2 - ввод кода слова (если это главное слово - 1, если вспомогательное - 2);

Insert - клавиша ввода в базу данных выбранного слова;

Kelete - клавиша уничтожения записи из базы данных;

kbbtn - клавиша возврата в главное меню;

Helpbtn - клавиша вызывающая информацию о данной форме;

Если в опции FILE главного меню выбирется опция CLKATA, то тогда появляется форма позволяющая модифицировать файл CLKATA. Данная форма показана на следующем рисунке:

При заполнении базы данных в “Position” записывается местоположение вводимого слова, само слово должно бсть записано в “Meaning”, формат информационного слова пишется в “Format”, если запись информационного слова в CLKATA содержит несколько строк, то номер каждой строки должен бить записан в “Line num-

ber”.

“Choice - 1” - позволяет определить функциональное назначение слова (является ли вводимое слово параметром, кодом или под-кодом CLKATA т. д.).

“Choice - 2” - определяет принадлежность слова к конкретной группе операторов (напр. к группе геометрических операторов, операторов движения и т. д.).

“Choice - 3” - определяющее принадлежность вводимого слова к группе главных или вспомогательных слов

кК - клавиша фиксирующая введенный код или под-код.

Insert - клавиша ввода записи CLKATA в базу данных CLKATA.

Cancel - клавиша позволяющая вернуться в главное меню.

Если в опции FILE главного меню выбирется опция KEVICE, то тогда появляется форма позволяющая редактировать существующее УЧПУ или вводить данные о новом УЧПУ. Форма модификации УЧПУ показанна на следующем рисунке:

Form4

Word

Modal - 1; Modalles - 0;

Format

Принадлежность:

Choice - 1

Geometric operators

Moving operators

Postprocessor operators

Additional operators

Choice - 2

Main word

Additional word

Kbnthfnehf

1. Shigley J.E. Mechanical Engineering Design/ 3-d ed., McGraw Hill Bok Company.

ჯგუფის შესახებ ...

ჯგუფი **ParametricCAD** შეიქმნა 1992 წელს სტუ-ს №381 ლაბორატორიაში. ჯგუფის ხელმძღვანელია ლაბორატორიის წამყვანი მეცნიერთანამშრომელი, ტ.მ.კ., დოც. ალ.შარმახანაშვილი. ჯგუფის სამუშაო ბირთვს ძირითადად ასპირანტები და მაგისტრები შეადგენენ:

1. ავთანდილ მამათავრიშვილი - ასპირანტი
2. ნელი დოლიძე - ასპირანტი
3. ლერი მეგრელიშვილი - ასპირანტი
4. ივა ზარაძე - მაგისტრი
5. ზვიად კაჭარავა - მაგისტრანტი
6. თეა წიკლაური - მაგისტრანტი
7. კობა კახეთელიძე - მაგისტრანტი
8. ეკა რაზმაძე - მაგისტრანტი
9. ელა გრიშიკაშვილი - მაგისტრანტი
10. პაატა მამფორია - მაგისტრანტი
11. ვიორჯი მარიაშიძე - მაგისტრანტი

ჯგუფში მიმდინარეობს საკვლევ-სამეცნიერო სამუშაოები მექანიკური დამუშავების ტექნოლოგიური პროცესების ავტომატიზებული სისტემების შექმნაზე. მომზადდა და წარმატებით დაცულ იქნა 6 სამაგისტრო დისერტაცია. ჯგუფში იქმნება სამომხმარებლო დანიშნულების პროგრამული პაკეტები. დამუშავდა სახარატო ოპერაციების ავტომატიზებული დაპროექტების სისტემა **Turbo T** (ინგლისურ და რუსულ ენოვანი ვერსიები), რომელიც მზადაა კომერციული ტირაჟირებისათვის.

ჯგუფის წევრები რეგულარულ მონაწილეობას იღებენ სამეცნიერო-ტექნიკურ კონფერენციებში. განსაკუთრებით საგულისხმოა საერთაშორისო სამეცნიერო ფორუმებში ჯგუფის წევრების მონაწილეობა - 1995 წელს, მეცნიერებათა აკადემიის მიერ გამართულ სიმპოზიუმში „პროექტების შემუშავებისა და კონვერსიის შესახებ“, 15-18 მაისი, ქ.თბილისი და 1998 წელს მანჩესტერის უნივერსიტეტში გამართულ სიმპოზიუმზე **“Tools and methods of Concurrent Engineering - TMCE'98”**, 12-15 მაისი, ქ.მანჩესტერი, ინგლისი.

დამყარებულია კავშირები, მონახულია საერთო სამეცნიერო ინტერესები და მიმდინარეობს ერთობლივი პროექტების მომზადება სხვადასხვა ქვეყნის სამეცნიერო ცენტრებთან - მანჩესტერის უნივერსიტეტი, ინგლისი (Prof.M.Latif), დელფის ტექნოლოგიური უნივერსიტეტი, ჰოლანდია (Prof.I.Horvath), დერბის უნივერსიტეტი, ინგლისი (Prof.T.Bendiab), ჰიუსტონის უნივერსიტეტი, აშშ (Prof.L.W.Chen), მოსკოვის ტექნიკური უნივერსიტეტი, რუსეთი (ГНУД=ДУИХТД) = ამ მხრივ აღსანიშნავია, რომ ჯგუფის ერთერთმა წევრმა (დ.მეგრელიშვილი) მიიღო მიწვევა გრძელვადიანი სამეცნიერო სტაჟირების ჩატარებაზე დერბის უნივერსიტეტის საინჟინრო სკოლაში (**School of Engineering**).

ParametricCAD ჯგუფი ყოველ წელს აწყობს სამეცნიერო კონფერენციებს სადაც ხდება ჯგუფში ჩატარებული სამუშაოების პრეზენტაცია, აპრობაცია და გამოცდილების ურთიერთგაზიარება.