

MISKOLCI EGYETEM
GÉPÉSZMÉRNÖKI KAR

NEHÉZ KÁROLY RÓBERT

A marás számítógépes szimulációja és optimalási kérdései

Doktori (Ph.D.) értekezés

Tudományos vezetők: Prof. Dr. Tóth Tibor, Dr. Csáki Tibor

Informatikai Intézet
Alkalmazott Informatikai Tanszéke
Miskolc, 2002

Jelölések jegyzéke és alapképletek

Rövidítés	A jellemző értelmezése	Mértékegység
D_c	Szerszámtátmérő <i>Cutting diameter</i>	mm
D_e	Adott fogásmélységhez tartozó maximális forgácsoló átmérő <i>Max cutting diameter at a specific depth</i>	mm
l_m	Forgácsolási úthossz <i>Machined length</i>	mm
d	Axiális fogásmélység <i>Axial cutting depth</i>	mm
w	Radiális fogásmélység <i>Radial cutting depth</i>	mm
v	Forgácsolási sebesség <i>Cutting speed</i>	$v = \frac{\pi \cdot D_c \cdot n}{1000}$ [m/min]
Q	Anyagleválasztási intenzitás <i>Metal removal rate</i>	$Q = \frac{d \cdot f \cdot w}{1000}$ [cm ³ /min]
R	Intenzitás dimenziójú változó	$\frac{\text{cm}^3}{\text{min}}$
t_g	Megmunkálási idő <i>Machining time</i>	$t_g = \frac{l_m}{f}$ [min]
z	Fogak száma <i>Number of teeth</i>	-
f_z	Fogankénti előtolás <i>Feed per tooth</i>	$f_z = \frac{f}{n \cdot z}$ [mm]
f_n	Fordulatonkénti előtolás <i>Feed per revolution</i>	$f_n = \frac{f}{n}$ [mm]

Rövidítés	A jellemző értelmezése	Mértékegység
k_c	Fajlagos forgácsoló erő <i>Specific cutting force</i>	$\left[\frac{N}{\text{mm}^2} \right]$
f	Előtolási sebesség <i>feedrate</i>	$f = \frac{f_z \cdot n \cdot z}{1000}$ [mm/min]
h_{ex}	Maximális forgácsvastagság <i>Max chip thickness</i>	[mm]
h_m	Átlagos forgács vastagság <i>Average chip thickness</i>	$h_m \approx f_z \cdot \sqrt{\frac{w}{D_c}}$ [mm]
n	Főorsó fordulatszám <i>Spindle speed</i>	$n = \frac{1000 v}{\pi \cdot D_c}$ [ford/min]
P_c	Forgácsolási teljesítmény <i>Cutting power net</i>	$P_c = \frac{d \cdot w \cdot f \cdot k_c}{60 \cdot 10^6}$ [kW]
κ_r	Főforgácsoló élszög <i>Major cutting edge angle</i>	fok
$R_{\max}$ $R_{\min}$	Makro paraméter korlátok a (Q,R) síkon <i>New macro parameter constraints in the plane (Q,R)</i>	$\frac{\text{cm}^3}{\text{min}}$
$Q_{\max}$ $Q_{\min}$	Intenzitás korlátok a (Q,R) síkon <i>Intensity constraints in the plane (Q,R)</i>	$\frac{\text{cm}^3}{\text{min}}$

Rövidítés	A jellemző értelmezése	Mértékegység
$v_{\max}$ $v_{\min}$	Forgácsolási sebesség korlátok a (Q,R) síkon <i>Cutting speed constraints in the plane (Q,R)</i>	$\frac{\text{m}}{\text{min}}$
T_a T_f	Éltartam korlátok a (Q,R) síkon <i>Tool life constraints in the plane (Q,R)</i>	min
δ_f	Szerszám legnagyobb megengedhető elhajlása <i>Maximum cutter deflection</i>	mm
$P_{\max}$	Maximális teljesítmény <i>Maximum power available</i>	kW
ϑ	Szerszám forgás szöge <i>Cutter rotation angle</i>	rad
$\vartheta(\vartheta, \varphi)$	Szerszám élelem forgásszöge <i>Cutting edge element rotation angle</i>	rad
φ	Szerszám élelem helyzeti szöge <i>Cutting edge element position angle</i>	rad
φ'	Szerszám élelem helyzeti szöge X'Y'Z' -ben <i>Transformed cutting edge element position angle in X'Y'Z' coordinate system</i>	rad
ϑ'	Szerszám élelem forgásszöge X'Y'Z' -ben <i>Transformed cutting edge element rotation angle in X'Y'Z' coordinate system</i>	rad

Rövidítés	A jellemző értelmezése	Mértékegység
ξ	Előtolási irány szöge <i>Feed inclination angle</i>	rad
β	Csavarvonal menet emelkedési szöge <i>Cutter helix angle</i>	rad
$h(\vartheta)$	Forgácsvastagság ujjmaró esetén <i>Chip thickness in case of flat-end cutter</i>	mm
$h(\vartheta, \varphi)$	Forgácsvastagság gömbvégű szerszám esetén <i>Chip thickness in case of ball-end cutter</i>	mm
dF_x dF_y	Sugárirányú és érintőirányú elemi forgácsolási erő <i>Radial and tangential elemental cutting force</i>	N
$K_T(\varphi)$ $K_R(\varphi)$	Sugárirányú és érintőirányú tapasztalati paraméter <i>Radial and tangential empirical cutting parameter</i>	-
m_T, m_R	Tapasztalati model paraméterek <i>Model parameters characterizing the workpiece material</i>	-
V_m	Leválasztott anyagtérfogat <i>The material volume to be removed</i>	cm^3
$\frac{\Delta \tau}{\tau}$	Relatív költségeltérés <i>Relative cost difference</i>	-
$\frac{\Delta T}{T_0}$	Relatív éltartameltérés <i>Relative tool life deviation</i>	-

Rövidítés	A jellemző értelmezése	Mértékegység
C_s	Komplex szerszám jellemző <i>Complex feature of the tool</i>	-
t_s	Szerszám költség ekvivalens idő <i>Tool cost equivalent time</i>	min
$C_v \ x_v \ y_v$ $z_v \ p_v \ w_v$	Éltartamegyenlet konstans kitévői <i>Empirical quasi-constants of Taylor equation</i>	-
t_{cs}	Egy él cseréjének átlagos ideje <i>The average time necessary for changing a tool edge</i>	min
K_{sb}	Szerszám alkalmazásával kapcsolatos teljes költség <i>The total cost connected with the use of tool</i>	Ft
K_Σ	Teljes költség <i>Total cost</i>	Ft
q	Tapasztalati konstans <i>Empirical constant</i>	$q = \frac{1}{m}$
γ_z	Hálófelbontás z-map esetében <i>z-map distance</i>	mm
ε_k	Kitüremkedési hiba <i>protrusion error</i>	mm
ε_f	Felületi hiba <i>surface error</i>	mm

Rövidítéslista

CIM	-	Computer Integrated Manufacturing - számítógéppel integrált gyártás
IMS	-	Intelligent Manufacturing System - intelligens gyártórendszer
MIS	-	Management Information System - vállalati információs rendszer
MRP	-	Manufacturing Resources Planning, Material Requirement Planning - gyártási, termelési erőforrások tervezése
FMS	-	Flexible Manufacturing System - rugalmas gyártórendszer
FMC	-	Flexible Manufacturing Cell - rugalmas gyártócella
MC	-	Machining Center - megmunkálóközpont
NC	-	Numerical Control - számjegyvezérlés
CNC	-	Computer Numerical Control - számítógépes számjegyvezérlés
CAD	-	CAD Computer Aided Design - számítógéppel segített termék és eszköztervezés
CAM	-	Computer Aided Manufacturing - számítógéppel segített gyártás
CAPP	-	Computer Aided Process Planning - számítógéppel segített folyamat tervezés
CAQ	-	Computer Aided Quality - minőség szabályozás
CAST	-	Computer Aided Storage and Transportation - számítógéppel segített tárolás és raktározás
MRR	-	Material Removal Rate - anyagleválasztási intenzitás
OpenGL	-	Open Graphics Library - "nyitott grafikus könyvtár" eredetileg a Silicon Graphics Inc. grafikus szabványa
API	-	Application Programming Interface
SSM	-	Sculptured Surface Machining - szoborszerű felületek megmunkálása
PIXEL	-	A képernyő egy pontja, legkisebb képi elem
DEXEL	-	Mélység elem - geometriai marás-szimuláció egyik módszere
Z-MAP	-	Z-térkép – a takarási feladat megoldására használt módszer
CSG	-	Constructive Solid Geometry
OOP	-	Objekum orientált programozás
MFC	-	Microsoft foundation classes - Microsoft alap osztálykészlet
STL	-	Standard template library - Szabványos C++ generikus osztályok gyűjteménye

Köszönetnyilvánítás

A disszertáció a Miskolci Egyetem Alkalmazott Informatikai és Szerszámgépek Tanszékén 1997-ben kezdett kutatómunkám eredményeit foglalja össze. A kutatás során többek támogatását élveztem. Köszönettel tartozom azoknak, akik a disszertáció megírásában segítségemre voltak. Hálás vagyok **Dr. Velezdi György** egyetemi adjunktus kollégámnak, aki először ösztönzött a számítógépes megmunkálás-szimuláció témájának kutatására és számos gyakorlati problémára hívta fel a figyelmem. Sokan nyújtottak segítséget azzal, hogy részben, vagy egészben átolvasták a disszertáció korai változatait, és építő javaslatokat tettek. Köszönet illeti **Dr. Dudás László**, **Dr. Erdélyi Ferenc** kollégáimat, akik ötleteik sokaságával és támogatásukkal segítették a kutatómunkámat. Köszönettel tartozom tudományos vezetőimnek, főként **Dr. Tóth Tibor** professzoromnak, aki először fordította a figyelmem az optimalizációs módszerek felé és precíz, tudatos kutatómunkára ösztönzött. Köszönet illeti **Dr. Csáki Tibort**, aki sok hasznos tanácsot nyújtott a disszertáció megírása során.

Végül szeretném megköszönni feleségem, családom és barátaim támogatását.

A marás számítógépes szimulációja és optimalizációs kérdései

Doktori (PhD) értekezés

írta

NEHÉZ Károly Róbert

Összefoglalás

A disszertáció első része a marási művelet számítógépes szimulációjával foglalkozik. Ebben a részben a szakirodalomból ismert modellekre alapozva egy olyan új szimulációs módszert mutatok be, amely az ismert eljárásokkal összehasonlítva több kedvező tulajdonsággal rendelkezik. Továbbá bemutatásra kerül egy szoftver prototípus is, amely az említett módszert felhasználva alkalmas a három- és öttengelyes marás geometriai szimulációjára, néhány minőségi jellemzőjének meghatározására. A disszertáció egy további fejezete a forgácsolási folyamatok anyagleválasztási-intenzitás alapú optimalizáló eljárásával foglalkozik. Ez a módszer egy sajátosan új megközelítésen alapszik, amelyet a Miskolci Egyetem Alkalmazott Informatikai Tanszékének kutatócsoportja, Tóth Tibor professzor vezetésével sokéves kutatómunka során fejlesztett ki. Kidolgoztam a nagyoló homlokmarásra alkalmazható optimalizációs modellt, megállapítottam a fontosabb korlátfeltételeket és egy minta adatbázist alakítottam ki a *Prototyp GmbH* német szerszámgyártó cég megmunkálási adatai alapján. A bemutatott eljárás jól alkalmazható a modern gépgyártástechnológia három hierarchiai szintjén, különböző teljesítőképességű alkalmazásokba ágyazva. Egyik alkalmazási példája maga a bemutatott marás-szimulátor szoftver, amely ez által nem csak a munkadarabból leválasztott anyagtérfogat közelítő számítására alkalmas, hanem az anyagleválasztás *off-line* szabályozására is. Külön fejezetben foglalkozom a megmunkáláskor fellépő erőket közelítő, a megmunkálás-szimulátorokban hatékonyan alkalmazható eljárásokkal. A szimulátor szoftver bemenő adatainak a *CAM* rendszerek által generált úgynevezett *CLData* fájl, ami tartalmazza az adott művelet, műveletelem geometriai és technológiai információit. A sikeres szimuláció után, a geometriai, minőségi ellenőrzést követően, a szoftver előállítja az NC program-kód fájlt, ami módosított technológiai paramétereket is tartalmazhat. Ezt az utólagos feldolgozó, átalakító folyamatot posztprocesszálnak nevezzük.

Az öttengelyes mozgásinformációkat is tartalmazó *CLData* fájlok posztprocesszálása nem végezhető el a konkrét gépkonfiguráció ismerete nélkül, mivel ekkor a posztprocesszálni folyamat nem csupán szintaktikai feldolgozást jelent, hanem figyelembe kell vennie az öt vezérelt tengely valós, térbeli elhelyezkedési viszonyait is. A szimulátor szoftver képes egy úgynevezett két körasztalos gépkiepités modellezésére, amely a Miskolci Egyetem Szerszámgépek Tanszékének műhelycsarnokában működik. A szoftverben alkalmazott eljárás jól általánosítható, a transzformációk, ütközés-vizsgálatok, linearizációs eljárások más gépkonfigurációk esetén is alkalmazhatók. A disszertációban bemutatott módszerek és modellek számos továbbfejlesztési irányt tártak fel, de nem hagytam figyelmen kívül ezek elvi és gyakorlati korlátait sem.

Tartalomjegyzék

1	Bevezetés.....	11
1.1	A kutatás célkitűzései	13
1.2	A kutatás módszere	13
1.3	A disszertáció szerkezete.....	13
2	Forgácsolási folyamatok geometriai szimulációja és ellenőrzése	15
2.1	A szimuláció és ellenőrzés fontossága	15
2.2	Történeti áttekintés	17
2.2.1	Hagyományos megközelítés	17
2.3	A 3 tengelyes marás formális megközelítése	17
2.4	Analitikus módszerek.....	18
2.5	Approximációs, közelítéssel eljárások	19
2.5.1	Diszkrét vektor eljárások	19
2.5.2	Z-map modell	20
2.5.3	Egyéb diszkrét eljárások	23
2.6	A szimulációs modellel szemben támasztott követelmények	24
2.7	Diszkrét modellek pontossági kérdései.....	26
2.8	Az új szimulációs módszer bemutatása	28
2.8.1	Az OpenGL szabvány.....	29
2.8.2	Megmunkálás-szimuláció P-buffer segítségével.....	31
2.8.3	A z-buffer pontossága.....	36
2.8.4	Az 5 tengelyes megmunkálás szimulációja	36
2.8.4.1	Transzformáció	36
2.8.4.2	Linearizáció	39
2.8.4.3	Szerszámgép konfiguráció modellezése és megjelenítése.....	40
2.8.4.4	Ütközés felügyelet, ütközés érzékelés	41
3	Számítógéppel segített folyamat tervezés (CAPP) szabadformájú felületek esetén	42
3.1	Folyamattervezési modellek.....	43
3.1.1	Szekvenciális folyamat tervezési modell.....	43
3.1.2	Hierarchikus folyamat tervezési modell.....	43
3.2	A technológiai paraméterek szerepe a folyamat tervezésben.....	44

4	Nagyoló homlokmarás költségoptimalizálása anyagleválasztási intenzitás segítségével.	46
4.1	Költségfüggvény forgácsoló megmunkálások esetén.....	46
4.2	Fajlagos költségekvalens idő.....	49
4.3	A korlátfüggvények	51
4.3.1	Éltartamkorlát	51
4.3.2	Teljesítménykorlát.....	52
4.3.3	Forgácsolási sebesség korlát.....	52
4.3.4	Szerszám megengedhető elhajlása	53
4.4	Tipikus döntési folyamatok a (Q, R, τ) állapotterben	53
4.4.1	Direkt feladat	53
4.4.1.1	Szabályos keresési tartomány	53
4.4.1.2	Magas éltartam alsókorlát.....	56
4.4.1.3	Alacsony teljesítménykorlát	57
4.4.2	Indirekt feladat.....	57
4.5	A módszer bemutatása egy konkrét példa alapján	59
4.6	Az éltartam szórás költségre befolyásolt hatása	63
4.6.1	A relatív költségeltérés levezetése.....	63
4.6.2	Egyszerűsítések szabályos keresési tartomány esetén.....	67
4.6.3	A hibavizsgálat összefoglalása	69
4.7	Minta – adatbázis létrehozása	70
5	Forgácsolási erő modellezése megmunkálás-szimulátorokban	73
5.1	Bevezetés.....	73
5.2	Marás	74
5.2.1	Térfogat modellek	76
5.2.2	Mechanikai modellek	78
5.2.3	A mechanikai modell megvalósítása	79
5.2.3.1	A pillanatnyi forgácsvastagság.....	79
5.2.3.2	Az érintkezési tartomány	80
5.2.4	A forgácsolóélre ható erő meghatározása.....	81
5.2.5	Az empirikus konstansok meghatározása.....	86
5.3	A biztonságos megmunkálási feltételek ellenőrzése	86
5.3.1	A fogakban ébredő feszültség	86
5.3.2	Szerszámelhajlás miatt keletkező felületi hiba.....	87
5.3.3	A módszer alkalmazása a saját fejlesztésű megmunkálás- szimulátorban.....	88
6	A prototípus megmunkálás-szimulátor szoftver bemutatása	89
6.1	A szimulátor főbb funkciói	90

6.2	A funkciók bemutatása példák alapján	92
6.3	A szimulátor sebességének összehasonlítása	98
7	Új tudományos eredmények	101
8	A tézisekhez kapcsolódó tudományos publikációk.....	104
9	Hivatkozások jegyzéke	107
10	Ábrajegyzék	115
	Első melléklet: Több-változós nemlineáris regresszió.....	117
	Második melléklet: Minta-adatbázis (elvi modell)	119

1 Bevezetés

A gyártási folyamat során, amíg az előgyártmányból késztermék lesz, a munkadarab alakjának és anyagtulajdonságainak megváltoztatásához több, különféle gyártási eljárást alkalmaznak. A legfontosabbak ezek közül a következők: öntés, képlékeny alakítás, forrasztás, hegesztés, *forgácsolás*, ragasztás, szerelés stb. A modern gyártástechnológiában a forgácsolási folyamat, mint alakítási eljárás, továbbra is nagy szerepet kap (a fém alkatrészek közel 80 százaléka ma is forgácsolással készül), mivel egyszerűsége és egyetemessége mellett, alkalmas magasabb pontossági követelmények gazdaságos feltételek melletti kielégítésére, nagy tömegszerűség mellett is. A forgácsolás eljárását Bali [1] a következőképpen definiálja: A *forgácsolás* a szilárd test alakját változtató olyan anyagszétválasztási eljárás, amelynek során a munkadarabról mechanikai úton, meghatározott eszközzel - a *szerszámmal* - anyagrészecskéket - *forgácsot* - választanak le. A forgácsolás és az egyéb alakadó eljárások szoros kölcsönhatásban vannak egymással, fejlődésük jelen pillanatban is párhuzamosan megy végbe.

A gépipari alkatrészek előállítása során, alakjuk és anyagtulajdonságaik szakaszosan változnak meg, tehát a folyamatok jellege nem folytonos, hanem diszkrét. A gyártási folyamatban geometriailag jól definiált munkadarabok megmunkálása folyik, szekvenciálisan rendezett megmunkálási műveletek segítségével. A megmunkálási eljárás tervezésekor, más műszaki feladatokhoz hasonlóan, alapvető követelmény, hogy a gyártandó darabok minőségének biztosítása mellett a gyártási költségeket csökkentsük, a műszakilag lehetséges megoldások halmazából megkeressük a leg gazdaságosabbat.

Forgácsolási folyamatok geometriai és fizikai vizsgálatát célszerű megfelelően megválasztott modellek segítségével végezni. A Természettudományi Lexikon [2] a modellt a következőképpen definiálja:

modell: Bonyolult rendszerek egyszerűsített, minden részletben áttekinthető, gyakorlatilag megvalósított vagy szemléletesen elképzelt, arányosan lekicsinyített vagy felnagyított, matematikailag szabatosan leírható, idealizált mása, amely többé-kevésbé helyesen szemlélteti a vizsgált rendszer vagy folyamat sajátosságait. A modellalkotásnál tudatában kell lenni annak, hogy a modell nem azonos a vizsgált rendszerrel vagy folyamattal, és nem tükrözi maradéktalanul az összes tulajdonságait. A helyesen megalkotott modell mégis magán viseli az objektív világban meglévő rendszer vagy a lejátszódó folyamat fontos ismérveit, és így alkalmas a döntő törvényszerűségek feltárására és szemléltetésére. Modellek létrehozásának fő célja az, hogy segítségével szimulálni tudjuk bonyolult rendszerek jövőbeli viselkedését. A [21] osztályozása szerint léteznek *képszerű*, *analóg*, *szimbolikus* és *kombinált* modellek.

A *képszerű modellek*, statikus módon, képszerűen jelenítik meg a valóság modellezendő területét. Az *analóg modellek* a valóság és a modelljellemzők közötti jól definiált szoros

kapcsolatokat érzékeltetik. A *szimbolikus modellek* a matematikai és logikai szimbólumok segítségével írják le a vizsgált folyamat egyes tényezőit.

A mi szempontunkból a *kombinált modellek* a legfontosabbak, amelyek részben az analóg, részben a szimbolikus modellek tulajdonságait ötvözik. A számítógéppel integrált gyártás területén legtöbbször ilyen modelleket alkalmaznak. A kombinált modellekkel szemben két egymással ellentétes követelményt támasztunk: (1) minél hűebben tükrözzék a valóságot; (2) minél egyszerűbbek legyenek. A valósághűsége való törekvés növeli a számítási igényt, mivel bonyolítja a matematikai struktúrát. Az egyszerűsége való törekvés a módszer számítási igényét csökkenti, és az implementálhatóságot egyszerűsíti. Meg kell találni a megfelelő kompromisszumot a modellek tervezésekor, figyelembe véve a szoftvertechnológia adott fejlettségi szintjét.

Algoritmuson egy feladat számítógépen való megoldására szolgáló eljárást értünk. Tehát a megalkotott, megsejtett modelleket csak jól kidolgozott algoritmusok segítségével lehet tesztelni, hatékonyságát megállapítani.

A disszertáció marási műveletek geometriai szimulációjával, ellenőrzésével, optimalási kérdéseivel foglalkozik. Ismeretes, hogy a diszkrét technológiai folyamatok tervezésének különböző hierarchiai szintjein számos optimalási feladat jelenik meg. Az optimalás fogalma általánosságban azt jelenti, hogy olyan intézkedéseket keresünk egy rendszerben, amellyel valamilyen maximális hatást érhetünk el. Az optimális megoldás mindig függ az elérendő céltól, ezért hatását mindig az értékelési kritériumhoz kell viszonyítani.

Az optimális megoldás megkeresését a modellek bonyolultsága jelentősen megnehezíti. Az optimaló és szimulációs algoritmusok nagy részénél nem lehet figyelmen kívül hagyni a heurisztikus megoldásokat és gondolkodásmódot. Amint a [22] munka megállapítja: „szinte nélkülözhetetlen a tanulás és újratanulás és a heurisztika. ... a problémakör megoldásának kulcsa a hierarchikus szemléletmód, az eltalált strukturálás és a tipizálás.”

A heurisztikus gondolkodásmód az adott probléma megoldását egy terv konstruálása útján keresi. Egy heurisztikus megoldásmód általában egyszerűbb, mint a matematikai modellekre épülő, mindig tartalmaz szubjektív tényezőket, ezért jóságát a gyakorlati alkalmazhatóságon lehet lemérni. A disszertáció keretén belül ezért fektettem nagy hangsúlyt a kidolgozott módszerek, algoritmusok gyakorlati implementálására, tesztelésére.

A szakirodalom szerint az optimális megoldás megkeresésének másik ismert módszere az utólagos szabályozás [4]. Egyes forgácsolási eljárások (különösen szabadformájú felületek megmunkálása) esetén, az egyes műveletelemek elvégzése utáni szabálytalan ráhagyási alakzatok, sztochasztikus jellegű, dinamikus hatásokat gyakorolnak a forgácsolószerszámmra. Ezek a negatív hatások sok esetben növekvő költséget, növekvő átfutási időket, és nem utolsósorban minőségromlást okozhatnak. A műveletelemek geometriai és forgácsolási erő szimulációja segítségével a dinamikus hatások utólagosan szabályozhatóak.

A diszkrét gyártási folyamatok tervezési folyamata öt egymástól jól elkülöníthető hierarchiai szinten történik. A hierarchia legmagasabb szintjén a műveletsor vagy műveletcsoportból álló halmaz helyezkedik el. Ennek egy eleme, egy lehetséges műveleti sorrend. Egy adott műveleti sorrend műveletekből áll (2. szint), a műveletek részműveletekből

(3. szint), a részműveletek műveletelemekből (4. szint), a műveletelemek pedig mozzanatokból állnak (5.szint). A disszertáció az utolsó két (4. és 5.) hierarchiai szinten elvégezhető technológiai optimalizálási és geometriai szimulációs lehetőségeket tárgyalja, a felsőbb szintek lényegesen eltérő feladataival nem foglalkozik.

1.1 A kutatás célkitűzései

Megvizsgálva a különböző ismert módszereket a megmunkálások számítógéppel segített geometriai szimulációja témakörben, arra a következtetésre jutottam, hogy hardveres gyorsítás alkalmazásával jobb minőségű és lényegesen gyorsabb szimulációs algoritmust lehet kifejleszteni. A hardveres gyorsítást, nem valamilyen célhardver, hanem szokványos, átlagosnak mondható grafikus hardver alkalmazásával céloztam meg.

A tanszékünkön már hosszú évek óta folyó intenzitás alapú optimalizálási módszerek felkeltették az érdeklődésemet, és célul tűztam ki ennek továbbfejlesztését homlokmarásra. A szimulátorok képesek az egyes műveletelemek elvégzése során maradó szabálytalan ráhagyási alakzatok közelítéses számítására, ez lehetőséget teremt a forgácsleválasztási intenzitásszámításra, amely a szimulátorokat, posztprocesszorokat képessé teheti a megmunkálási paraméterek off-line szabályozására, a gyártási költségek és a termelékenység optimalizálására.

1.2 A kutatás módszere

Kutatómunkám során a problémafeltevés, irodalomkutatás, megoldási módszerek keresése, implementálás, tesztelés, kiértékelés lépésekből álló ciklikus folyamatot követtem.

Az irodalomkutatás során, alapvetően két tématerületet tekintettem át. Az egyik terület a forgácsolási folyamatok intenzitás alapú optimalizálásával foglalkozik. A másik pedig a számítógépes megmunkálás geometriai szimuláció és ellenőrzés tématerülete, ezen belül a hangsúlyt a marás specifikus problémákra helyeztem.

Az elkészült kísérleti, prototípus szoftverek tesztelése, értékelése számos olyan új problémát vetettek fel, melyek megoldása újabb módszerek, technikák felhasználását, fejlesztését tették szükségessé. A kutatás során arra törekedtem, hogy minél több modern szoftvertechnológiai eszközt elsajátítsak és alkalmazzak.

1.3 A disszertáció szerkezete

A disszertáció következő fejezetében a forgácsolási folyamat geometriai szimulációs modelljeinek történeti áttekintését és alkalmazási lehetőségeit mutatom be intelligens

gyártórendszerekben. Ezen kívül bemutatásra kerül egy új marás-szimulációs módszer, amely főként sebességi és virtuális gyártórendszerben való alkalmazhatósági előnyökkel rendelkezik az eddig ismert módszerekkel összehasonlítva.

A harmadik és negyedik fejezetekben egy új anyagleválasztási intenzitáson alapuló külső optimalizálási módszert mutatok be homlokmarás esetén. Ez az eljárás elsősorban a gyártórendszer belső hierarchiájában, a műveletelem szintjén jelenik meg, mint optimalizálási feladat. Továbbá, egy gyakorlati példán keresztül bemutatásra kerül a módszer alkalmazhatósági lehetősége CAPP rendszerekben és megmunkálás-szimulátorokban. Megvizsgálom a módszer alkalmazhatósági határait, a szerszám éltartamnak a gyakorlatban adódó sztochasztikus viselkedése figyelembevételével.

Az ötödik fejezetben először a megmunkáláskor fellépő erők modellezési lehetőségeinek történeti áttekintésével, és hatásával foglalkozom, majd megvizsgálom a bemutatott módszerek implementálási lehetőségeit az új marás-szimulációs módszerrel. Ebben a fejezetben a gyártórendszer legalsó hierarchiai szintjén, a mozzanat szintjén jelentkező optimalizálási lehetőségekről lesz szó. Elsősorban a hagyományos ujjmaró és gömbvégű szerszámokkal foglalkozom.

A hatodik fejezet a saját fejlesztésű prototípus megmunkálás-szimulációs szoftver bemutatásával foglalkozik. A szoftver a második fejezetben bemutatott új eljárás demonstrációját szolgálja.

A hetedik fejezet tartalmazza az új tudományos eredményeket megfogalmazó téziseket, majd a nyolcadikban található a tézisekhez kapcsolódó tudományos publikációk listája, a kilencedik fejezet a felhasznált legfontosabb szakirodalmi hivatkozásokat tartalmazza.

A mellékletekben bemutatásra kerül a maráshoz használható mintaadatbázis felépítésénél használt matematikai modell és egy minta adatbázis.

2 Forgácsolási folyamatok geometriai szimulációja és ellenőrzése

Az intelligens alkatrészgyártás egyik legfontosabb törekvése az, hogy a szerszámgépek működése közben minimálisra csökkentsék az operátori felügyelet szintjét. Ideális esetben az NC gépek automatikus üzemmódban működnek és ennek eléréséhez az NC program szimulációs és ellenőrző szoftverek felhasználása alapvető fontosságú. Egyszerű, hasáb alakú alkatrészek esetén elegendő a megmunkáló gépen egyszeri próbafuttatást alkalmazni (programbelövési fázis), majd a megfelelő darabszámú alkatrészeket sorban legyártani. Szabadformájú (free-form) felületeket tartalmazó alkatrészekből általában csak néhány darab gyártása szükséges, ezért a próbafuttatás végrehajtási ideje gyakran nagyon magas a teljes sorozat átfutási idejéhez képest. Az előgyártmányok ára gyakran olyan magas, hogy már komoly költségtöbbletet okozhat egy esetleges helyrehozhatatlan selejt alkatrész előállítás. A programbelövési idő csökkentésének legkézenfekvőbb módszere valamilyen NC program ellenőrző szimulációs módszer alkalmazása. A modern CAM alkalmazások fontos és elengedhetetlen komponense lett az NC program ellenőrző szimulátor.

2.1 A szimuláció és ellenőrzés fontossága

A szimuláció, a mérnöki munka minden területén fontos szerepet kap. A szimuláció a valós világ minél tökéletesebb, teljesebb utánzására törekszik és szimulációs modelleket hoz létre. Az ellenőrző módszerek, a szimulált modellek alapján az elfogadhatóság kritériumait vizsgálják. A forgácsolási folyamatok esetén alkalmazott szimulációs és ellenőrző eljárások ideális esetben a következő kérdésekre keresik a válaszokat:

1. Az adott műveletelem alkalmazásakor a megmunkált felület megfelel-e az előírt tűrés követelményeinek?
2. Van-e valamilyen nemkívánatos mellékhatás, ütközés? Például: a munkadarabfelfogó készülék ütközik-e a szerszámmal, vagy a szerszámbefogó készülék ütközik-e a kész felülettel?
3. Mekkora az adott műveletelem termelékenysége, költsége?
4. A forgácsoláskor fellépő erők nem okoznak-e szerszámtörést vagy túl nagy szerszámelhajlást, esetleg gyors szerszámkopást?

Ezekre a kérdésekre kielégítő választ akkor kaphatunk, ha megvizsgáljuk, és osztályozzuk a megmunkálás közben lehetséges hibákat. A 1. ábrán az NC megmunkáláskor keletkező hibák csoportosítása látható. Az egyes hibatípusok a különböző tevékenységekhez egyértelműen hozzákapcsolhatóak.

A mai korszerű, kereskedelmi forgalomban kapható NC program szimulátor és ellenőrző szoftverek általában csak az NC program szintű hibák feltárását és automatikus vagy interaktív korrigálását teszik lehetővé. Ez abból adódik, hogy a NC program nézőpontjából lehetetlen a felsőbb hierarchiai szintek vizsgálata, hogyha például nincs semmilyen információnk a tényleges megmunkáló gépről, gyártócelláról, arról a tágabb környezetről, ahol a megmunkálás valójában végbemegy. A NC kód generáló posztprocesszorok csak a legalapvetőbb geometriai, technológiai információkat tárolják az egyes gépekről, vezérlésekről. A szerszámpálya generátor szoftverek az egyes megmunkáló műveleteket általános gépekre hozzák létre (processzor-posztprocesszor elv). A hagyományos megmunkálások esetében ez a módszer jól bevált és a mai napig elfogadott.

Folyamat	- szerszámtörés - szerszámkopás - szerszámkihajlás - munkadarab elhajlás - nem megfelelő forgácsleválás
Operátor	- hibás munkadarab felfogás - hibás szerszámtár szerszámozás - hibás szerszám geometriai korrekciók beállítása - hibás override funkció
Megmunkáló gép	- gép tartóelemek deformációja - szervóvezérlési hiba - főorsó pontatlanság - hőokozta deformációk - hűtőfolyadék fogyás
NC program	- munkadarab alámetszése - szerszámütközés a munkadarab, szerszám befogó készülékekkel - hiányos anyageltávolítás - posztprocesszási hiba - a megmunkált felület maradóborda magassága túl nagy - szerszám pillanatnyi megállásából adódó felülethiba - hibás főorsó fordulatszám - hibás előtolási sebesség - túl nagy fogásmélység és fogásszélesség

1. ábra: Az NC megmunkáláskor keletkező hibák csoportosítása az egyes szinteken [26]

Vannak azonban olyan modern megmunkálási eljárások, amelyek technológiai tervezésekor is szükséges a gépkonfiguráció geometriai kialakításának ismerete. A szerszámpálya generátor szoftverek ilyen esetben is dolgozhatnak hagyományos geometriai rendszerben (munkadarabhoz kötött rendszerben), de a posztprocesszási fázishoz már nélkülözhetetlen a valóságos elrendezés ismerete. Az öttengelyes megmunkálási módszerek ebbe a kategóriába sorolhatóak. Általánosságban igaz az a kijelentés, hogy az öttengelyes megmunkáláshoz szükséges a működéséhez elengedhetetlen a tényleges gépkonfiguráció geometriai elrendezésének ismerete. Ezzel a többletinformációval, a megmunkáló gép szintjén jelentkező hibák nagyrésze is korrigálható. A hőokozta deformációk és a főorsó pontatlanságának vizsgálatával, korrekciójával számos publikáció foglalkozik [109,110].

2.2 Történeti áttekintés

2.2.1 Hagyományos megközelítés

Számtalan eljárást dolgoztak ki az NC programok ellenőrzéséhez. Ezek közül a legegyszerűbb, a számítógép nélküli hagyományos módszer, az NC program próbafuttatása. Ilyenkor a munkadarabot fa, műfa vagy műanyag nyersdarab helyettesíti vagy nyersdarab nélküli próbafuttatást is szokás végezni. A módszer hátránya az időigényesség, költségesség és ezek mellett az, hogy az 1. ábrán bemutatott hibatípusok nagy részének felismerése nem lehetséges, de ettől eltekintve, még ma is sok helyen alkalmazzák egyszerűsége miatt.

A számítógépes módszerek alkalmazása ezen a területen is egyre szélesebb körben terjed. A számítógéppel segített NC program szimuláció legegyszerűbb formája nem más, mint a szerszámpálya vonalakkal való ábrázolása előre megadott nézőpontokból a számítógép képernyőjén. Ennek egy fejlettebb megvalósítása még ma is megtalálható minden ismert CAM szoftverben. Az ellenőrzés úgy történik, hogy a szoftver, a szerszámot animálva mozgatja az NC soroknak megfelelő pálya mentén, közben a kész munkadarab felülete is látható, tetszőleges nézőpontba forgatható, skálázható. Anyagleválasztás szimuláció nincs, a szerszám elmozdulásait színes vonalakkal kötik össze. Az ilyen vizuális ellenőrzéssel minden nagyobb hiba felismerhető. Ennek az alapszámítógépes módszernek egy modern adaptációja figyelhető meg [42]-ben a virtuális valóság (virtual reality) technológia felhasználásával, mivel ezen egyszerű szimulációs módszer legnagyobb előnye a gyors implementálhatóság és elenyésző hardverteljesítmény-igény. Összefoglalásként kijelenthetjük: a vizuális megfigyelésen alapuló módszerek általában pontatlanok és időigényesek.

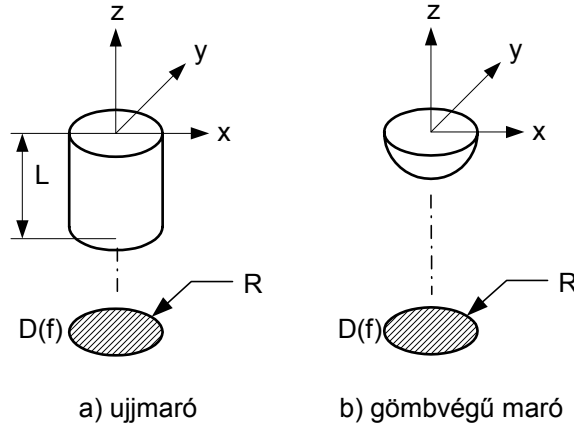
A disszertáció célkitűzései között az anyagleválasztás alapú szimulációs-módszerek kutatása szerepel, ezért a továbbiakban ennek modellezési kérdéseivel foglalkozom.

2.3 A 3 tengelyes marás formális megközelítése

A pontosabb számítógépes módszerek áttekintése előtt vizsgáljuk meg a probléma elvi hátterét. A háromtengelyes marás formális leírása a következő [92]: Legyen $\mathbf{M}$ olyan egyenes szakaszok halmaza, amely a szerszám egy kitüntetett (referencia vagy programozott) pontjának helyzeteit tartalmazza a háromdimenziós térben, ezt a halmazt más néven szerszámpályának nevezzük. Legyen f olyan függvény, ami a szerszám származtató felületét reprezentálja és az origó körüli szimmetrikus körlap alakú tartományon értelmezett. Példaként definiálhatjuk a lapos (1) és gömb végű (2) szerszámok származtató felületeinek $f(x,y)$ kétváltozós függvényeit:

$$f(x,y) := -L, \quad x,y \in D(f) := \{(x,y) \mid x^2 + y^2 \leq R^2\}, \quad (1)$$

$$f(x,y) := -\sqrt{R^2 - x^2 - y^2}, \quad x,y \in D(f) := \{(x,y) \mid x^2 + y^2 \leq R^2\}. \quad (2)$$



2. ábra: az ujjmaró és gömbvégű marószerszámok származtató felületeinek $f(x,y)$ kétváltozós függvényei.

A képletben szereplő L és R a szerszám jellemző méretei, hossza és sugara. Mindkét esetben az origót használják referencia pontként. Tegyük fel, hogy a nyersdarab egy téglatest, amely az $x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}$ érték hatással adható meg. Legyen adott $s(x,y)$ függvénnyel az előgyártmány felülete is, ami legegyszerűbb esetben $s(x,y) = z_{max} = const$. Keressük a megmunkált felületet reprezentáló $F(x,y)$ függvényt:

$$F(x,y) := \min\{z_{max}, \min\{z \mid z := s(x',y') + f(x-x',y-y'), (x-x',y-y') \in D(f), z \geq z_{min}, (x',y',s(x',y')) \in M\}\}. \quad (3)$$

A referencia pontot végigvezetve a szerszámpályán, az f függvény egy térfogatot határoz meg, és e térrész és az előgyártmány különbsége adja a megmunkált munkadarab térfogatot. A megmunkálás geometriai szimulációja nem más, mint az $F(x,y)$ függvény vizualizációs folyamata. Ez a folyamat, a már megvalósított eljárások alapján két fő csoportba sorolható: analitikus és közelítési módszerek.

2.4 Analitikus módszerek

Az $F(x,y)$ analitikus módszerrel történő, a halmazelméleten alapuló számítási módszerét Gossard és Tsuchiya tette közzé 1978-ban [72]. A szilárdtest modellezés felhasználása Voelcker nevéhez fűződik, aki a PADL (CSG) szilárdtest modellező rendszer NC geometria

szimulációs és ellenőrző célokra való felhasználási lehetőségeit kutatta [73]. Ezen módszerek esetében a szerszám, mint szilárd test halad végig a szerszámpályán, és adott mintavételezési pillanatokban a CSG rendszer elvégez egy test-test kivonás műveletet, ezzel szimulálva az anyageltávolítást. Ennek eléréséhez a rendszernek felület-felület metszések sorát kell végrehajtania, ami nagyon időigényes feladat. Jelentős gyorsulást ért el Kawashima [84], egy úgynevezett "Graftree" módszer alkalmazásával növelte a hatékonyságot. A módszer jól ötvözi az Octal-tree [33] és a CSG módszer előnyeit: az oktálfa csúcsaiba CSG elemeket ágyaztak a pontosság és minőség növelése érdekében, az CSG alkalmazása pedig megszünteti a csipkézettség okozta minőségromlást. A szilárdtest modellező megközelítés legnagyobb hátránya általában az, hogy $O(N^4)$ a számítási igénye [73], ahol az N a szerszámpálya szakaszainak számát jelenti. Komplex munkadarabok esetén a szerszámpálya több tízezer szakaszból állhat, ami a CSG alapú megközelítés alkalmazhatóságát teljesen lehetetlenné teszi (még a mai átlagos számítási sebességek mellett is). A kereskedelmi forgalomban kapható NC szimulátorok közül az NSEE használja ezt az eljárást (Az NSEE a Microcompatibles cég terméke).

2.5 Approximációs, közelítéssel eljárások

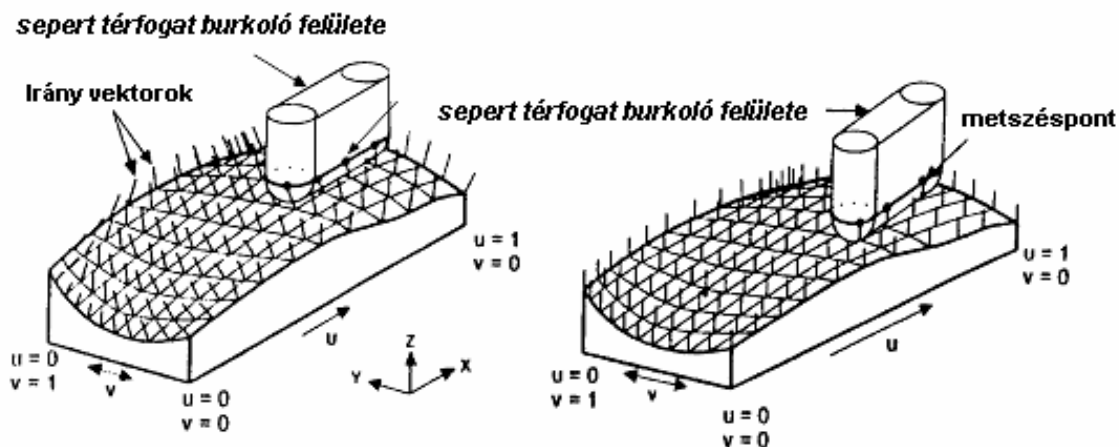
A közelítéssel eljárások két csoportba sorolhatók: diszkrét vektoros és z-térképezés (z-map) módszerek. A nagyobb hangsúlyt a z-map eljárások tanulmányozására érdemes helyezni, mert napjainkban túlnyomóan ezek bizonyultak a leghatékonyabbnak, mivel képesek kielégíteni a fejezet elején említett kérdéses formában megfogalmazott követelményeket.

2.5.1 Diszkrét vektor eljárások

A szerszámpálya mentén mozgó szerszám származtató felülete egy térfogatot, térrészt határoz meg (swept volume) amely burkolófelületének meghatározása a cél. A módszer a mozgó szerszám burkolófelületének és a megmunkálandó felület normálvektorainak elmetszésén alapul [74] (3.ábra). Ez az eljárás elsősorban a megmunkálás pontosságának megállapításában nyújt jelentős segítséget. A gyártandó felület modellen, előre meghatározott tartományban, megadott lépésközzel képeznek a munkadarab felületén pontokat, és ezekből induló külső normálvektorokat. A megmunkálást ellenőrző folyamat során a művelet pontossága megállapítható a pont-vektor párok (*rays*) és a szerszám mozgása közben létrejövő úgynevezett seprét térfogat burkolófelülete közötti távolság meghatározásával. Az ellenőrzési folyamat három részfeladatot foglal magába: diszkretizáció, lokalizáció, metszésképzés. A diszkretizáció folyamatában a rendszer a munkadarabmodell felületén megfelelő sűrűséggel képezi a pontokat és normálvektorokat. A lokalizációs folyamat kiválasztja ezek részhalmazát az aktuális szerszámhelyezethez, a metszésképzés megállapítja a távolságot a szerszám és munkadarab között, ahol a vektorok hossza negatív, ott alámetszés történt, ahol pedig pozitív,

ott el nem távolított anyag maradt. A marási hiba grafikus ábrázolása színekkel történik a felületmodellen. A [74]-ban részletes metsző algoritmus található véletlen helyzetű henger és előre megadott vektorsokaság között. Oliver [81] egy ettől eltérő, de elvében hasonló módszert dolgozott ki. Az ő megközelítése a megmunkált felület képtérben reprezentált modelljéből indul ki. A felhasználó tetszőlegesen választhat egy felületrészt, amely a vizsgált tárgyfelület lesz. Ennél az eljárásnál a szimuláció a képtérben megy végbe: a képernyő minden egyes pontjából (pixelből) a képernyő síkjára merőlegesen egyenes szakaszokat (vektorokat) indít a tárgyfelület felé és megállapítja a metszéspontokat. Ez a metszésponthalmaz lesz a felület approximációja, és a képernyő síkjából induló vektorokon lehet alkalmazni a metsző algoritmust.

A vektormetsző módszerek hátrányaként a nagy memóriaigényt és a kis számítási sebességet lehet megemlíteni. Ennek ellensúlyozására több olyan gyorsító eljárás is született, amely a felület és vektorok metszési sebességének növelésével, és a vektorok számának csökkentésével foglalkoznak [83,97]. További lényeges hátránya az, hogy a leválasztott anyagtérfogat meghatározása nem lehetséges, mivel ez a technika szorosan kötődik a felületmodellhez. A módszer segítségével előállított megmunkált felület vizuális megjelenítésében a mai gyors grafikus hardverek és az OpenGL technológia nagymértékben segíthet.



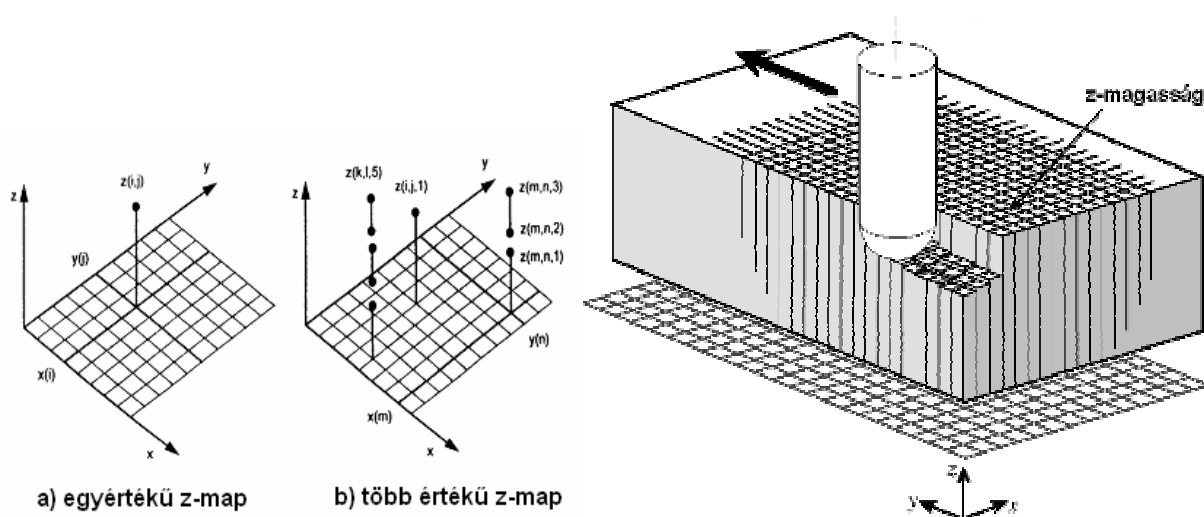
3. ábra: a diszkrét vektorok iránya lehet a felületi normálisokkal (lásd bal oldali ábra) vagy a szerszám tengelyével (jobb oldali ábra) párhuzamos [78,80].

2.5.2 Z-map modell

A z-map módszer napjainkban a leggyakrabban alkalmazott geometriai szimulációs eljárás. Az egyik legelterjedtebbnek mondható Vericut szoftver is ezt a modellezési módot alkalmazza [107]. A módszer alapja először Anderson 1978-as publikációjában található meg [71]. A cikk olyan NC programok generálási problémáival foglalkozik, amelyek mentesek a

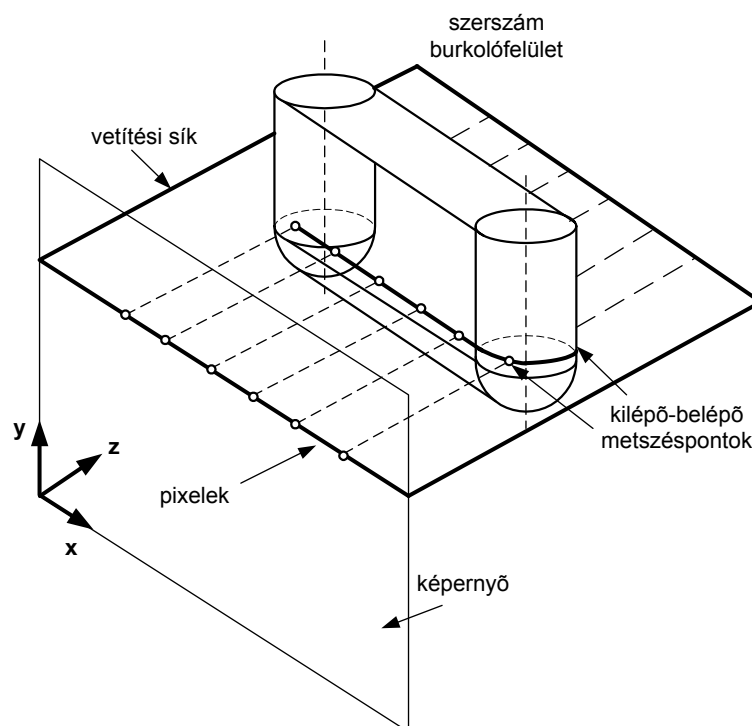
szerszámbefogó és munkadarab ütközéseitől. Az ütközés meghatározásához a szerszámot és a munkadarabot négyzet alakú hasábokkal reprezentálta, és hasáb-hasábmetszésekkel alkalmazva állította elő a megmunkált felületet. Háromdimenziós hisztogrammnak nevezte el az így kialakult struktúrát és mátrix segítségével tárolta. A módszerére nem z-map néven hivatkozott, de mégis ezt tekinthetjük az eljárás alapművének. A műveletigény $O(N)$, tehát a szerszámpálya szakaszok száma és a műveletigény között lineáris a kapcsolat, ami lehetővé teszi a nagy számú szakaszokból álló szerszámpályák szimulációját és vizsgálatát is. (4. ábra)

A z-map módszer számos későbbi független kutató publikációjában jelent meg eltérő megvalósításokban [75,77]. A módszer előnye magában foglalja a könnyebb implementálhatóságot, gyors szimulációt, a forgácseltávolítás közelítő számítását, és egyszerű összehasonlíthatóságot biztosít a megmunkált és tervezett felületek között.



4. ábra: a legszélesebb körben alkalmazott z-map modell elvi alapját mutatja be. A baloldalon az egy és többértékű z térkép elvét mutatja, a jobboldali pedig a vizualizáció legegyszerűbb módját hasábokkal.

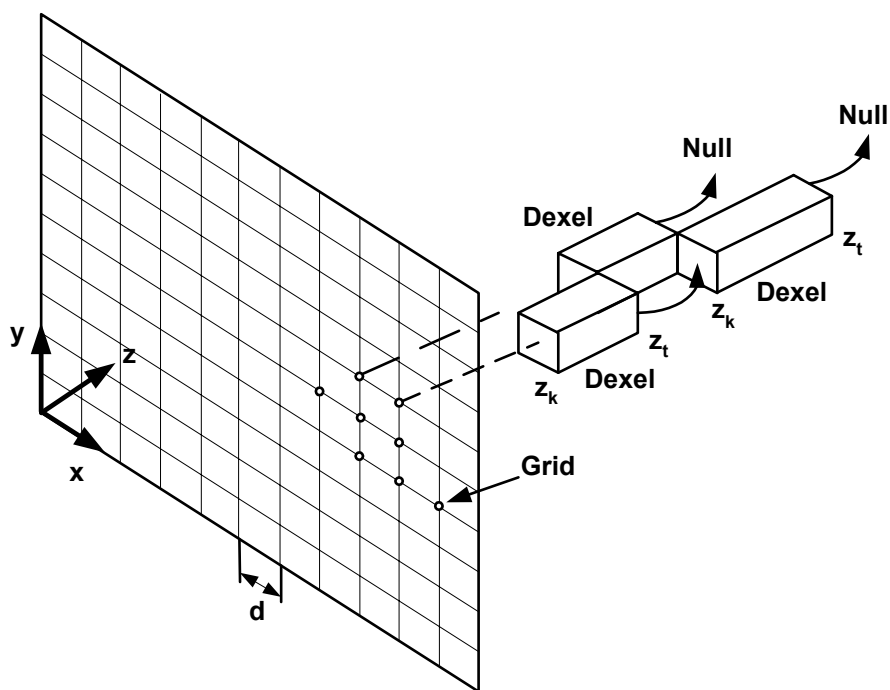
Érdemes megfigyelni két eltérő, de mégis hasonló megvalósítást, ami jól mutatja a módszer sokszínűségét és ötletgazdagságát. Atherton [77] által javasolt megvalósítás a legegyszerűbb. (lásd 5. ábra) A képernyő minden pixelére merőlegesen a vetítősugarak és a szerszám burkolófelület metszéspontjait kell kiszámítani, amennyiben van metszéspont, le kell tárolni a ki és belépő pontokat egyaránt.



5. ábra: Atherton [77] által javasolt megvalósítás a legegyszerűbb: a képernyő minden pixelére merőlegesen a vetítősugarak és a szerszám burkolófelület metszéspontjait kell kiszámítani, amennyiben van metszéspont, le kell tárolni a ki és belépő pontokat egyaránt.

Van Hook [76] és Huang [97] módszere esetén a szimuláció egyaránt a képtérben megy végbe, egy kiterjesztett z-map módszer alkalmazásával (lásd 6.ábra), ami lehetővé teszi az 5 tengelyes szerszám pálya szimulációt is. Bevezette a *dexel* ('mélység elem' - depth element) fogalmát, minden *dexel*-t meghatároz a hozzá tartozó z_k (közeli) illetve egy z_t (távoli) metszéspontja. A számítógép képernyőjén előre megadott pixelekre merőleges egyenes szakaszokat képez, és kiszámítja a dexel-térben tárolt szerszám burkolófelületével való be-, illetve kilépő metszéspontokat, és láncolt listára fűzi őket.

A z-map módszerek pontossága egyaránt függ a dexelek sűrűségétől és a z irány megválasztásától. A dexelek számának növelése négyzetesen lassítja a számítási sebességet, és növeli a memóriaigényt. A dexelek elrendezése az esetek többségében egyenközü.



6. ábra: Van Hook [76] kiterjesztett z-map módszere, ahol a szimuláció a képtérben megy végbe, a dexelet (depth element) láncolt listában tárolják. Minden dexelt a hozzá tartozó z_k (közeli) illetve z_t (távoli) metszéspontjaival lehet megadni.

Fontosnak tartom Glaeser [95] úgynevezett „ Γ -buffer” módszerét kiemelni. Ez az eljárás némileg ötvözi az analitikus és diszkrét megközelítést, mivel a szerszám által seprt térfogat burkolófelületét differenciál geometriai eszközökkel számítja, a munkadarabot kiterjesztett z-buffer-, a szerszámot pedig poligonhalmaz reprezentálja.

Speciális forgácsolási eljárásokhoz speciális módszereket fejlesztettek ki. Yang és Lee [86] szikraforgácsoláshoz olyan dexel változatot hozott létre (R-map), ahol a dexelet kiinduló pontjai egy forgástengelyen helyezkednek el, és tömör hengert reprezentálnak.

Szabadon felvehető görbevonallú koordináta-rendszer R koordináta-irányát alkalmazza a [88,89] szerinti módszer felülettel burkolt felület meghatározására, elsősorban konjugált fogfelületek előállítására. A módszer a leseteinek tekinthetők az előbb említett hengerkoordinátás R bufferes

módszer, valamint a Descart-koordináta-rendszert alkalmazó z-bufferes megoldás is. A módszer általánosságáért azonban számítási idővel kell fizetni.

2.5.3 Egyéb diszkrét eljárások

Léteznek olyan diszkrét eljárások is, amelyek nem sorolhatóak sem a diszkrét vektor és sem a z-map eljárások közé. Érdekes módszer ezek közül a [99]-ban ismertetett „kockacukor”

módszer. Annak ellenére, hogy a jelzett cikk 1993-ban jelent meg, számos megvalósított szimulátor alkalmazza, például a Heidenhain 417-es vezérlésben is valószínűleg ezt a módszert alkalmazzák. Ebben a modellben a munkadarab és a szerszám egyaránt kockákkal reprezentált, és háromdimenziós mátrixban tárolt. Nagyolás szimulációra tökéletesen alkalmas és gyors, de simítás esetén már nem hatékony, főleg a rossz minősége és a nagy memóriaigénye miatt. A kirajzolási idő jelentősen lecsökkenhet, ha kihasználjuk az izometrikus vetítés sajátosságait. Léteznek teljesen Octal-tree alapú modellek is, amelyek a kockacukor módszer memóriaigényén próbálnak csökkenteni.

Egy másik módszert dolgozott ki az Alkalmazott Informatikai Tanszék 2002-ben végzett egyik hallgatója, egy OpenGL alapú stencil bufferes eljárást [30], amely az analitikus és diszkrét módszerek egyfajta ötvözésén alapul. Kihasznlta a z-buffer azon tulajdonságát, hogy segítségével egyszerű CSG műveletek hajthatók végre. A nagyolást meglepően jó minőségben szimulálja, de a nagy méretű szerszámpályák esetén nem lehet alkalmazni, az $O(n^4)$ műveletigénye miatt.

A következő fejezetben összefoglalom, hogy az itt bemutatott módszerek alapján, milyen követelményrendszert lehet felállítani a szimulációs eljárásokkal szemben.

2.6 A szimulációs modellel szemben támasztott követelmények

Az előzőek alapján arra a következtetésre jutottam, hogy az NC megmunkálás-szimulációhoz a *megmunkálás közben folyamatosan változó munkadarab*-, a *kész munkadarab*-, és a *szerszám származtató felülete által seprert térfogat*, mint három geometriai entitás modellezésére van szükség. A kész munkadarab geometriai modellezésére azért van szükség, mert ennek hiányában nem lehet semmilyen ellenőrzési módszerről beszélni a későbbiekben.

Öttengelyes megmunkálás esetén negyedikként a *szerszámgép* geometriai modellezése is szükséges. A munkadarab geometria dinamikusan változik (térfogata csökken), miközben a szerszám a megadott szerszámpálya mentén mozog, és anyagot távolít el belőle. Olyan algoritmus kifejlesztése a cél, amely gyorsan és pontosan tudja modellezni ezt az anyageltávolítási folyamatot. Pontosságában el kell érnie a 1:10000 arányt, ami nélkülözhetetlen a geometriai ellenőrzés szempontjából. 1 méter hosszú munkadarab esetén is lehetővé kell tenni az egy százados (0.01 mm) ellenőrzési pontosságot. Már említettem, hogy a szerszámpályák gyakran több mint tízezer szerszámpozíciót is tartalmaznak, ezért elengedhetetlenül fontos a gyors algoritmusok kifejlesztése és alkalmazása.

A modellek követelményrendszere annak a függvényében változik, hogy milyen típusú szimulációs és ellenőrzési feladatot kell végrehajtani. A különböző műveletelemek szimulációjának eltérő követelményei lehetnek: például a nagyolás és félsimítás esetén az anyagleválasztás intenzitása nagy, és a fellépő forgácsolási erők is jelentősek. A leválasztott anyagtérfogat pontos becslése elengedhetetlen feltétele az erők pontos meghatározásának. Ebben a fázisban a ráhagyási alakzat térfogatának minimalizálása is fontos szerepet kaphat, de kevés figyelmet kell fordítani az alámetszések ellenőrzésére.

A simító műveletek ettől némileg eltérő követelményrendszert igényelnek: itt általában az anyagleválasztás intenzitása és a fellépő forgácsolási erők jelentősen csökkennek. Nagyobb hangsúlyt kap a felületi minőség ellenőrzése, mivel ebben a szakaszban a szerszám és tervezett-felület érintkezésben vannak. A pontossági követelmények jelentősége itt a legnagyobb: pontos képet kell kapni a felületi hibákról, meg kell állapítani, hol vannak esetleges alámetszések. A csökkenő anyagleválasztási intenzitás és erők ellenére a szerszám elhajlása komoly felületi hibát okozhat.

Az NC program ellenőrző eljárások közül a következő négyet célszerű alkalmazni a szimulátorokban: (1) a kész felület vizuális ellenőrzése a teljes NC program szimulált futtatása után (2) a szerszám egyes elmozdulásainak megfigyelése, grafikus animáció segítségével, (3) a megmunkált felület és a tervezett felület összehasonlítása (4) a megmunkálás közben fellépő erők és szerszám elhajlások összehasonlítása a megengedett értékekkel.

Mind a négy eljárásnál szükség van nézőpont független munkadarab vizsgálatra. Ez azt jelenti, hogy kiegészítő követelményként azt is el kell várni egy algoritmustól, hogy képes legyen dinamikusan változtatni a nézőpontot a szimuláció fent említett eljárásokban, elősegítve ezzel a valódi munkatér szimulációt. Tapasztalataim azt mutatják, hogy a kereskedelmi forgalomban kapható szimulátorok ezt csak részben vagy egyáltalán nem tudják teljesíteni. Ez a kiegészítő követelmény azért fontos, mert ennek hiányában nem lehet megvalósítani a valós idejű 5 tengelyes anyagleválasztás szimulációt és grafikus animációt, mégpedig azért nem, mert a forgó és billenő asztalokon elhelyezett munkadarab megmunkálás közben dinamikusan változtatja a pozícióját.

Összefoglalásként a következő táblázat tartalmazza a 8 legismertebb szimulációs módszert, kilencedikként kiemelve a saját fejlesztésű „p-buffer” eljárással, amely a későbbiekben kerül bemutatásra. A módszerek osztályozása a szerszám, munkadarab, seprő térfogó, alámetszés figyelés, nézőpont függőség alapján történt.

	munkadarab	szerszám	alámetszés s figyelés	szerszám alakja	nézőpont függő	sepert térfogat	tengelyek száma
„Dexel” [76, 77]	dexel		Nincs	tetszőleges	igen	képtérben	3,5
„Extended dexel” [97]	dexel		Van	tetszőleges	nem	dexeltérben	3,5
„Lawn moving” [78]	háló normál vektorokkal	poligonális	Van	gömbvégű és lapos végű	nem	poligon	3,5
„Graftree” [84]	octree, CSG		Van	tetszőleges	nem	CSG	5
„G-buffer” [82]	kiterjesztett Z-Buffer		Van	tetszőleges	igen	G-buffer	3
„wire EDM”[86]	R-buffer		Van	elektróda (huzal)	nem	poligon	4
„moving solid” [73,87]	analitikus		Nincs	tetszőleges	nem	analitikus 4d függvény	3,5
„Γ- buffer”[95]	Γ-buffer	analitikus	Van	tetszőleges	nem	Γ-buffer poligonális	3,5
„P-buffer”	z-buffer	poligonális	Van	tetszőleges	nem	z-buffer	3,5

7. ábra: a nyolc bemutatott megmunkálás-szimulációs eljárás, kilencedikként kiemelve a saját fejlesztésű p-buffer eljárással. A módszerek osztályozása a szerszám, munkadarab, sepert térfogat, alámetszés figyelés, nézőpont függőség alapján.

2.7 Diszkrét modellek pontossági kérdései

Mielőtt rátérnék az új módszer bemutatására, fontosnak tartom a pontossági analízist. Ez a rész arra a kérdésre keresi a választ, hogy milyen pontosságot lehet elérni diszkrét modellek alkalmazásával. Z-map modellezés esetén a felületi hibákat csak előre megadott diszkrét helyeken tudjuk vizsgálni. A tervezett felületet a z-map hálópontjaival mintavételezzük, és minden pontban összehasonlítjuk a szimulált megmunkált felület ugyanezen pontokban vett mintavételezett értékeivel. A 8/a ábra egyenközű (dx) mintavételezéssel diszkretizált felületet ábrázol. Ha a felület meredeksége nagy, akkor a mintavételezés hibája is nagyobb ($b > a$). Meredek felületeken tehát kisebb ellenőrzési pontosságot lehet biztosítani. Arra fogjuk keresni a választ, milyen sűrűn kell felvenni a mintavételezési pontokat, hogy a legnagyobb észlelhető bemetszési hiba (kitüremkedési hiba) ε_k legyen (8/b ábra). A vizsgálódásunkat kizárólag gömb végű szerszámokra korlátozzuk. Gömb és sík elmetszése esetén a következő legegyszerűbb geometriai elrendezést vizsgáljuk (8/c ábra). A szerszám sugara r , adott kitüremkedési hiba

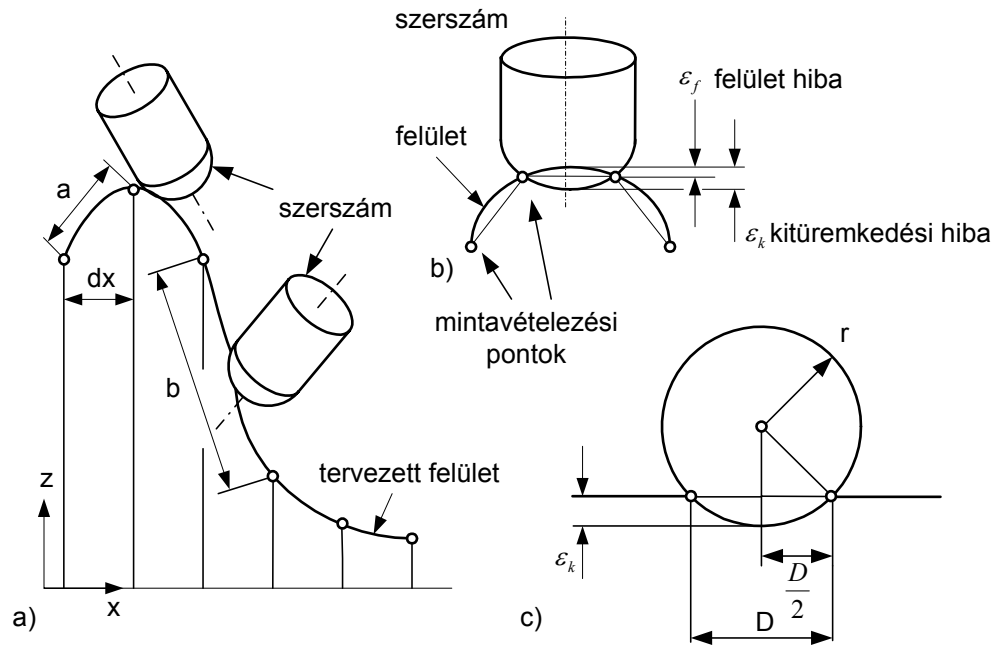
esetén a bemetszés húr hossza D . A húr hossz kifejezhető a következő képlettel (8/c ábra alapján):

$$D = 2\sqrt{2r \varepsilon_k - \varepsilon_k^2} \approx 2\sqrt{2r \varepsilon_k} . \quad [\text{mm}] \quad (4)$$

A 9. ábra a D sugarú körbe beírható egyenlő oldalú háromszög L oldalhosszának számítását mutatja. Helyettesítsük be az előző képletbe:

$$L = \frac{\sqrt{3}}{2} D = \sqrt{6r \varepsilon_k}, \quad \varepsilon_k = \frac{L^2}{6r} . \quad [\text{mm}] \quad (5)$$

Adott L és r esetén a legnagyobb bemetszési hiba a fenti képlet alapján könnyen számítható.



8. ábra: a diszkrét módszerek pontossági vizsgálatához szükséges modellek.

Ujjmaró és lekerekített marók esetén természetesen más képleteket kell alkalmazni [81]. A z-map alkalmazása esetén a kiszámolt L távolságot egy konstanssal és a felületi normális z irányú komponensével n_z kell módosítani [26]. Ebben az esetben a pontok távolsága, azaz a hálófelbontás:

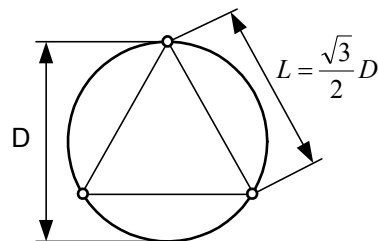
$$\gamma_z = 0.707 n_z L . \quad [\text{mm}] \quad (6)$$

Visszahelyettesítve (5)-be azt kapjuk:

$$\varepsilon_k = \frac{\gamma_z^2}{3r n_z^2} \quad [\text{mm}] \quad (7)$$

A kereskedelmi forgalomban kapható szimulátorok nagy részénél nem állítható a pontosság. A háló tipikus felbontása 512 x 512. A [26] szerinti számítás szerint ez a felbontás egy 250 mm hosszú munkadarab esetén 50 mm átmérőjű szerszám alkalmazásánál 0.003 mm pontosságot tesz lehetővé, de 10 mm átmérőjű szerszám esetén csak 1.6 mm (a számítások $n_z = 0.1$ esetén érvényesek).

Ettől függetlenül a z-map modell jól alkalmazható, de használatánál oda kell figyelni a pontossági kérdésekre, ugyanúgy, mint minden más közelítési eljárás esetén.

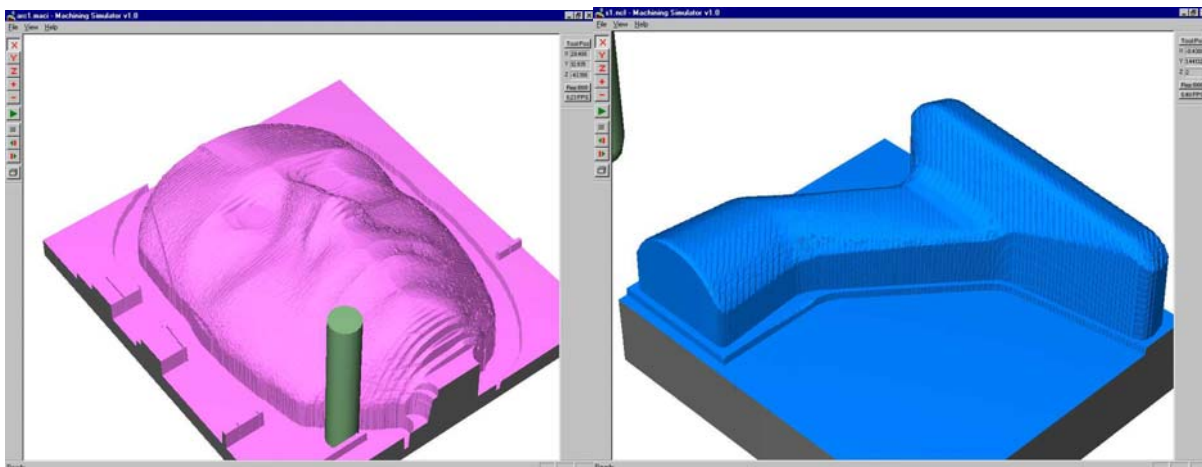


9. ábra: D sugarú körbe beírható egyenlő oldalú háromszög, L oldalhossza jelenti majd a mintavételezési pontok távolságát.

2.8 Az új szimulációs módszer bemutatása

A kutatás folyamán mindig az egyedi megvalósításoktól, prototípusoktól haladtam az általános elvek felé, az előző részben a követelményrendszert is egyfajta összegzésként, általánosításként mutattam be a szimulációs eljárások történeti áttekintése után. Az itt bemutatandó szoftver, egyszerű parancssoros 3 tengelyes posztprocesszor alkalmazásként indult 1995-ben, egy tudományos diákköri munka keretében, amit diplomamunkámban [25] fejlesztettem tovább 5 tengelyes megmunkálások esetére is. Ekkor merült fel először az igény a munkatér, és az anyagleválasztási folyamat valós idejű szimulációjára. Ez a követelmény akkor még túl bonyolultnak bizonyult, az akkor rendelkezésemre álló hardveren és szoftverfejlesztői tudásom alapján. Ekkor kezdtem el a már ismert módszerek irodalmának gyűjtését, implementációját, tesztelését, a megvalósítás során tapasztalt hátrányok és az előnyök elemzését. Elsőként a [99]-ben ismertetett „kockacukor” (voxel) módszer megvalósítását tűztem ki célul, a megmunkálás grafikus megjelenítésnél már sikeresen alkalmaztam az OpenGL rendszert. A már említett gyenge megjelenítési minőség miatt a továbbfejlesztést elvettem, más lehetőségeket kerestem. Következő lépésben a [84]-hez hasonló *octal-tree* módszerek implementációs lehetőségeivel foglalkoztam, és megvalósítottam egy teljesen *octree* alapú saját modellt. A megvalósított modelljeim nem voltak alkalmasak a szabadformájú

(free-form) felületeket is tartalmazó megmunkált alkatrészek modellezésére, a simítási művelet rossz grafikus megjeleníthetősége miatt. A „ Γ -buffer” [95] részbeni megvalósítása után arra következtetésre jutottam, hogy a felállított követelményrendszert kielégítő eljárásnak a z-map modellen kell alapulnia.



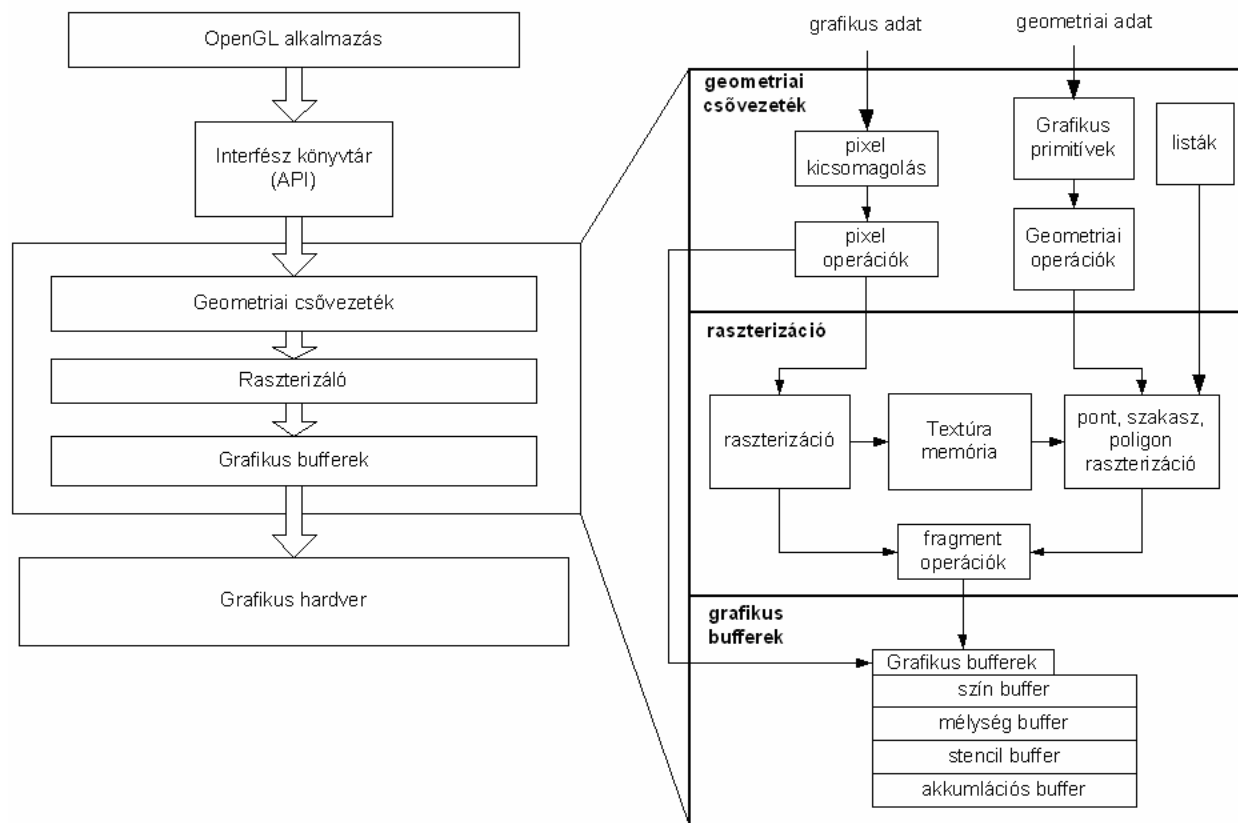
10. ábra: a „ Γ -buffer” [95] eljárás részbeni megvalósításával készült szimulátor. A megmunkált felületek valósághűek, de az anyagleválasztás szimuláció meglehetősen lassú. A jobb oldali alkatrész szerszámpályáját a Pro/Engineer szoftver generálta, a bal oldalit pedig egy magyar fejlesztésű CADtoMill elnevezésű szoftver. Az OpenGL könyvtár alkalmazása jó minőségű megjelenítést tesz lehetővé.

A 10. ábrán látható a Γ -buffer módszer alkalmazásával készült szimulátor. Legnagyobb hátrányként a lassú anyagleválasztás szimulációt kell megemlítenem.

Az ismert módszerek implementálása nyomán arra a következtetésre jutottam, hogy egyszerre kell növelni a szimulátor grafikus megjelenítési minőségét és a teljesítményt, e kettős cél megvalósítása érdekében keresni kell egy olyan API-t (Application Programming Interface) amely megfelelő hardveres gyorsítással teszi lehetővé az anyagleválasztás szimulációt és dinamikus, nézőpont független animált megjelenítést. Erre a célra az OpenGL szabvány modern implementációit tartom a legalkalmasabbnak. A következő fejezet rövid áttekintést ad az OpenGL rendszerről, majd megfogalmazza azon alaptulajdonságait, amelyek felismerése után, a fent említett kettős cél hatékonyan megvalósítható.

2.8.1 Az OpenGL szabvány

A grafikus API egy réteget képvisel a grafikus alkalmazás és a grafikus hardver között. Ezen keresztül közvetlenül elérhetőek a grafikus hardver erőforrásai és szolgáltatásai.



11. ábra: az OpenGL egyszerűsített architektúrája, amelyet a ma elérhető csaknem minden grafikus megjelenítő eszköz hardver szinten támogat.

Az OpenGL rendszert eredetileg a *Silicon Graphics Inc.* fejlesztette ki Unix alapú számítógép hálózatra (11. ábra). Az OpenGL, egy hardverfüggetlen programozási felület biztosító háromdimenziós grafikus rendszer. A legrészletesebb dokumentációk, példák gyűjteményét [35]-ben találhatjuk meg. A legismertebb és legtöbbször hivatkozott programozási útmutató az Addison Wesley kiadó gondozásában jelent meg [43].

Az OpenGL API segítségével geometriai vagy grafikus adatokat tudunk a csővezeték (*pipeline*) számára bementként megadni, amit a rendszer a raszterizáció során pixelekre bont, majd ezekhez színértékeket, textúra elemeket rendel és továbbítja a grafikus bufferekbe, majd a képernyőn bittérkép formájában jelenít meg. Vizsgáljuk meg a 11. ábrán bemutatott architektúrát abból a szempontból, hogy hogyan lehet anyagleválasztás szimulációra alkalmazni. Ennek megvalósításához, a következő lehetőségek nyújtanak segítséget:

1. A geometriai adatokat listákba lehet szervezni, ezeket a rendszer csoportosan, jelentős sebességnövekedés mellett kezeli.
2. A rendszer megteremti annak a lehetőségét, hogy a bufferek tartalmát kimásolhatjuk a memóriába, majd megváltoztatva azok tartalmát, visszamásolhatjuk a grafikus hardver memóriájába.

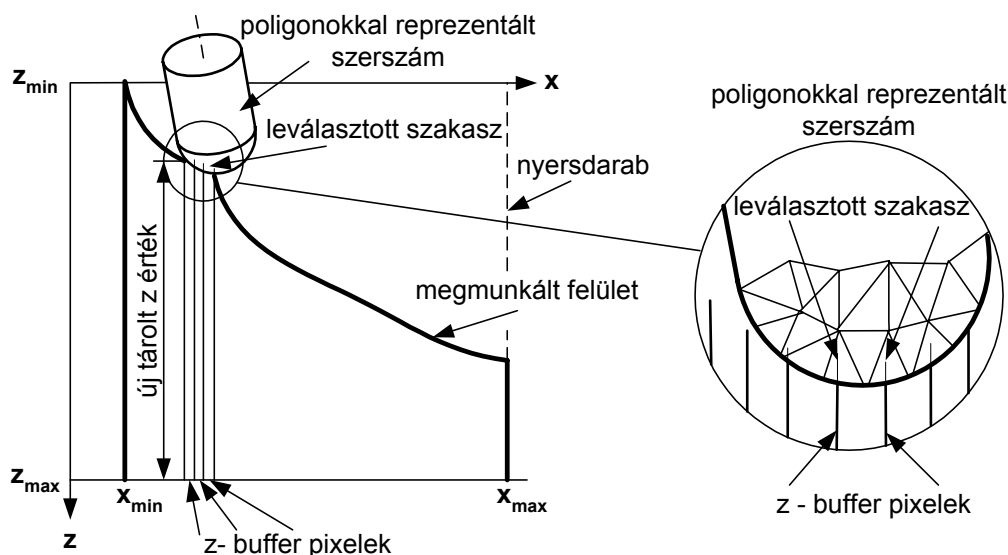
3. A rendszer állapotgépként működik, műveletei rugalmasan testre szabhatók, finoman hangolhatók. Ezzel a lehetőséggel mód nyílik a mélység buffer anyagleválasztás számító motorként való alkalmazására.
4. Az OpenGL egy nyitott rendszer, ezzel lehetővé teszi önmaga bővítését kiterjesztett funkciók segítségével. Így (hardver specifikus) új lehetőségek is alkalmazhatók a meglévő funkciók mellett.

Ez a négy tulajdonság alapozta meg az új szimulációs módszer kifejlesztését. Manapság az NVIDIA cég által kifejlesztett *GeForce* processzorok tartalmazzák az egyik leggyorsabb hardverben implementált grafikus csővezetékét. Az NVIDIA 4. pontban említett szabványos bővítmények egész sorát fejlesztette ki, legutóbbi fejlesztésük pedig a programozható grafikus processzor, amely új távlatokat nyitott az OpenGL fejlesztőknek [36]. A következő rész az új p-bufferes szimulációs módszerrel és implementációs lehetőségeivel foglalkozik.

2.8.2 Megmunkálás-szimuláció P-buffer segítségével

A z-buffer algoritmust a 70-es évek közepétől alkalmazzák a háromdimenziós grafikában, bár eleinte a nagy tárigénye miatt háttérbe szorult. A [33] meghatározása szerint: "A *z-buffer algoritmus* a takarási feladatot az egyes pixelekre oldja meg, oly módon, hogy minden pixelre megkeresi azt a poligont, amelynek a pixeleken keresztül látható pontjának a z koordinátája minimális." Ezt a keresést azzal támogatja, hogy minden pixelhez letárol egy aktuális mélység értéket. A poligonokat egyenként dolgozza fel és meghatározza a poligonok vetületén belül eső pixeleket. Inicializáláskor a z-buffer minden pixelében a tárolható legnagyobb érték van (∞ -nel történő inicializálás). A rajzolási folyamat során a poligonok vetületi képe jelenik meg a szín bufferben, de ugyanakkor pixelenkénti z -koordinátákat a z-buffer tárolja. Ha $p(i,j)$ pixelhez letárolt z koordináta *kisebb*, mint az aktuális poligon $p(i,j)$ pixelre eső z -koordinátája, akkor felülírjuk a letárolt z -koordinátát, mivel ez azt jelenti, hogy az aktuális poligon ezen $p(i,j)$ pixele nincs takarásban (közelebb esik a képernyő síkjához, mint a letárolt érték). Amennyiben z-buffer felülírás történik, akkor természetesen a képernyő megfelelő pixelének színe is módosul.

A szimuláció szempontjából fontos felismerés, hogy ha a z-buffert az előzőektől eltérően inicializáljuk, és az összehasonlító operátort megváltoztatjuk, akkor ez lehetőséget teremt a marás geometriai szimulációjához. Ezt könnyen be lehet látni a 12. ábra alapján: először inicializáljuk a buffert nullákkal ($z_{min} = 0$), állítsuk elő a szerszám poligonokkal közelített reprezentációját, minden poligon végezzük el a z-buffer műveletet olyan módon, hogy csak akkor írjuk felül a z - koordinátát, hogyha az aktuális pixel z -koordináta nagyobb, mint a tárolt. Ekkor a szerszám aktuális helyzetének „lenyomatát” kapjuk a bufferben.



12. ábra: a z-buffert $z_{min} = 0$ értékekkel inicializáljuk és akkor tároljuk le az aktuális z-koordináta értéket a bufferben, ha az nagyobb, mint a tárolt buffer érték. Ha a bemenő poligonok egy szerszámot reprezentálnak, akkor a szerszám az aktuális pozícióban egy „lenyomatot” hagy a bufferben.

Ha a CLData fájl minden szerszámhelyzetében elvégezzük ezt a műveletet, akkor a megmunkált felület „megjelenik” a z-bufferben.

Az átalakított z-buffer működés az OpenGL API segítségével könnyen implementálható, a felhasználó által nem látható „hátsó” színbufferben (*back buffer*). A módszernek az a hátránya, hogy a z-buffer mérete nem lehet nagyobb, mint az éppen aktív képernyőablak mérete. Bár a gyakorlat azt mutatja, hogy a szimulátorok általában az 512x512-es buffer felbontást használják. (Egy manapság szokványos 1024x768 pixel felbontású képernyőn realizálható egy 800x600-as buffer.)

Az NVIDIA által kifejlesztett egyik új lehetőség az úgynevezett p-buffer [37] kiterjesztés, ami lehetővé teszi a felbontás független mélység bufferek alkalmazását. (32 Mbyte videokártya memória esetén a p-buffer mérete 2048x2048 pixel.) Az új szimulációs módszer a hagyományos, p-buffert nem tartalmazó videó hardvereken is alkalmazható, a már említett felbontás megkötéssel.

A z-buffernek a megszokottól eltérő alkalmazása sok más kutatót is foglalkoztatott a közelmúltban. A teljesség igénye miatt említem meg, hogy még további nyolc alternatív alkalmazás létezik, amelyek [105,106]-ban összefoglalva megtalálhatók. Ezek az alkalmazások a főként a számítógépes grafika: képkompozíció, CSG rendering, árnyék térképek, térfogat vizualizáció, Voronoi diagrammok, szimmetria felismerés, objektum rekonstrukció, stb. témakörökbe tartoznak.

A továbbiakban a módszer néhány lényeges implementálási kérdésével, és a szimuláció folyamatának főbb lépéseivel foglalkozom. A modern objektum orientált paradigma alkalmazásával, a p-bufferes eljárás egyetlen osztályban is megvalósítható a következő tagfüggvények implementálásával:

- Lineáris koordináta transzformáció az $\{(x, y) \mid x \in [x_{\min}, x_{\max}], y \in [y_{\min}, y_{\max}]\}$ bázisról a $\{(u, v) \mid u \in [0, n_1], v \in [0, n_2]\}$ bázisra, és ennek inverze. Ez a transzformáció alakítja át az $x_{\min}, x_{\max}, y_{\min}, y_{\max}$ koordinátákkal jellemzett nyersdarabot, (és szerszámot, tervezett felületet) az u, v diszkrét bázisra. Az u, v diszkrét koordinátarendszerben ábrázolt geometria könnyedén kezelhető grafikus hardver segítségével.
- Megjelenítési lista alkalmazásával (lásd 13. ábra), az aktuális szerszám felületét reprezentáló háromszöghalmazt elő-feldolgozott formában, a videó hardverben kell tárolni. Szerszámcsere esetén új listát kell betölteni, vagy feldolgozni.
- Több nagyméretű kétdimenziós tömböt kell alkalmazni a szimuláció közbenső fázisainak és a tervezett felület tárolására.
- Inicializáló és hibakezelő tagfüggvények.
- Szükséges egy tagfüggvény, amely a szerszám geometria segítségével az aktív p-bufferben, (vagy z-bufferben, ha a hardver nem támogatja) végrehajtja az anyagleválasztás szimulációt. Az osztály erőssége az egyszerű implementációban rejlik, ezért bemutatom ennek a függvénynek azt C++ kódrészletét, amely magát az anyagleválasztást szimulálja (13. ábra).
- A p-buffer megjelenítésére szolgáló függvények.

```
//
// 3d translation for 3d machining
//
glTranslatef( fTransformXYToUV_X(ToolPosition.x),
              fTransformXYToUV_Y(ToolPosition.y),
              ToolPosition.z);

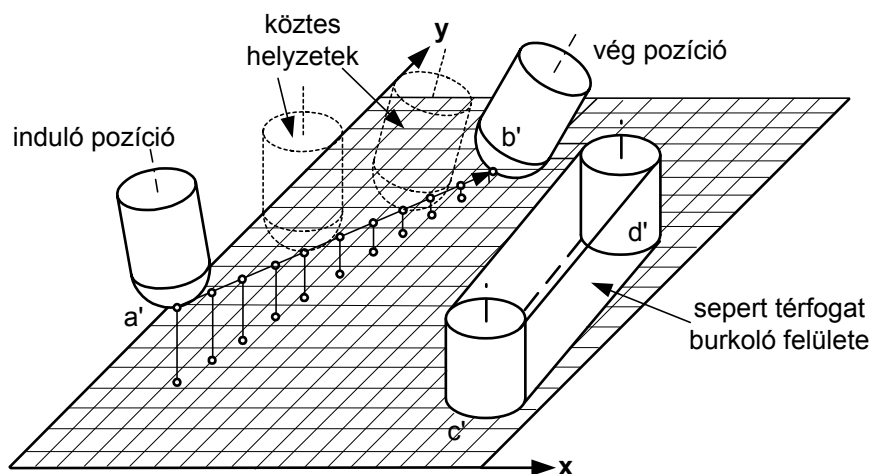
//
// scaling in UV space
//
glScalef(PBufferConstants.m_fXConstXYToUV, PBufferConstants.m_fYConstXYToUV, 1.0);
//
// two rotations for 5 axis positioning
//
glRotatef( ToolPosition.b, 1.0f, 0.0f, 0.0f);
glRotatef( ToolPosition.a, 0.0f, 1.0f, 0.0f);
//
// draw the current tool
//
glDrawArrays(GL_TRIANGLES, 0, m_vfToolRenderVertices.size() / 3);
```

13. ábra: az új p-buffer eljárás magja, a „tisztá” OpenGL alapú anyagleválasztás szimuláció (a soron következő szerszámhelyzetben). A vastagított függvények az OpenGL API függvényhívásai. A *m_vfToolRenderVertices* lista tartalmazza a szerszám geometriát. Az első két

függvényhívás a szerszámpozíció u, v koordináta-rendszerbe való transzformációt jelzi. A további két függvény a szerszám elfordulását jelöli, az esetleges negyedik és ötödik programozott tengely alkalmazásakor (5-tengelyes megmunkálás esetén). Az utolsó függvényhívás végzi el a videó hardverben az anyagleválasztás szimulációt.

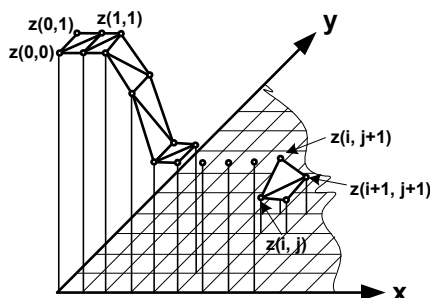
Az előzőekben bemutatott osztály alkalmazásával a marás-szimuláció a következőképpen történik:

1. Az inicializáláskor a tároló buffereket le kell nullázni. Be kell tölteni a szimulálni kívánt CLData fájlt, majd a kész felület geometriát. Az előre megadott pontosság figyelembevételével kell meghatározni az n_1 és n_2 konstansok értékét, amely a p-buffer méreteit jelenti.
2. Egy ideiglenes $n_1 \cdot n_2$ -es p-bufferben mintavételezni kell a már betöltött kész felület geometriát, és az eredményt egy $n_1 \cdot n_2$ mátrixban lehet letárolni.
3. A szerszám egy tetszőleges a és b pont közötti egyenes vonalú elmozdulását a p-bufferben úgy lehet szimulálni, hogy az ab szakaszt át kell transzformálni a diszkrét u, v tartományba és az így keletkezett ab' szakaszt fel kell osztani k darab részre, olyan módon, hogy az egyes szakaszpontok más-más pixelhez essenek (14. ábra). Ezekben a k pontokban kell elvégezni az anyagleválasztás szimulációt. 5-tengelyes megmunkálás esetén a köztes helyzetekben megfelelően elforgatva kell a szerszám poligonjait felhasználni. A tapasztalat azt mutatja, hogy a simításkor az ab' szakasz hossza olyan kicsi (a szerszámpálya generáláskor 0.02 mm-es tűrésmezővel dolgozva!), hogy átlagosan csak 1 pixel távolság van a' és b' között. E miatt simításkor nincs gyakorlati értelme a seprt térfogat burkolófelületét kiszámítani, mivel a felület kiszámítása, majd háromszögekkel közelítése több időt vesz igénybe, mint pixelenként újraszámolni a teljes szerszám-poligonokat [95].
4. Az anyagleválasztásnál fellépő forgácsolási erő és intenzitás közelítő számítását a köztes helyzetekben kell elvégezni, ezekkel a módszerekkel részletesen egy későbbi fejezetben foglalkozom.



14. ábra: marás-szimuláció a p-bufferben. Nagyolás esetén érdemes a szerszám seprt térfogatának burkolófelületét képezni, és egy lépésben szimulálni cd' szakasz megmunkálását, simítás esetén viszont köztes helyzeteket kell képezni ab' szakaszon, majd lépésenkénti szimuláció szükséges.

5. A felhasználó számára a p-buffer láthatatlan, ezért szükség van egy hatékony megjelenítő módszerre. A legegyszerűbb megoldást a p-buffer z magasság értékeinek háromszögekkel való közelítése adja (15.ábra), melynek előnye a gyors implementálhatóság. Hátránya az, hogy a háromszögek száma egy 512×512 -es buffer esetén több mint félmillió. A háromszögek számának csökkentésére léteznek rekurzív algoritmusok, a legjobb példa erre az [95]-ben bemutatott módszer. Általánosságban elmondható, hogy a mai modern grafikus hardver 100-200 ezer poligont (háromszöget) már körülbelül 10-szer is képes a képernyőre rajzolni másodpercenként, természetesen sebességre optimált kód esetén. Az implementálási tapasztalat azt mutatta, hogy a rekurzív módszereknél az említett legegyszerűbb eljárás (lásd 15.ábra) lényegesen gyorsabb és pontosabb. Érdemes alkalmazni a térinformatikából ismert LOD (level of detail) algoritmusokat is a gyors kirajzolás érdekében. A szimulátor szoftver 5 tengelyes megmunkálásnál körülbelül 15 képkocka per másodperc sebességű animációt biztosít, egy átlagos grafikus hardveren.



15. ábra: legegyszerűbb buffer-megjelenítési algoritmus, amely háromszögeket illeszt a végpontokra. A kirajzolandó háromszögek száma a felére csökkenthető, ha négyszögekkel dolgozunk.

2.8.3 A z-buffer pontossága

A napjainkban, kereskedelmi forgalomban kapható grafikus kártyák mindegyike tartalmaz hardveres z-buffer gyorsítást, a z-buffer mérete pixelenként 32 bit (4×8 bit). A 4×8 bit pontosság megfelel a szabványos *float* fix pontosságú lebegőpontos tárolási típusnak. 32 biten 6 tizedes jegy pontosságot lehet elérni (például: 0.000000 és 0.999999 intervallumban) akkor ez a pontosság a szimuláció szempontjából azt jelenti, hogy 1 méter magas alkatrész esetén is elvileg el lehet érni a $1\mu\text{m}$ pontosságot! Tehát e szerint, a z-buffer módszerrel messzemenően kielégíthetőek a pontossági követelmények.

2.8.4 Az 5 tengelyes megmunkálás szimulációja

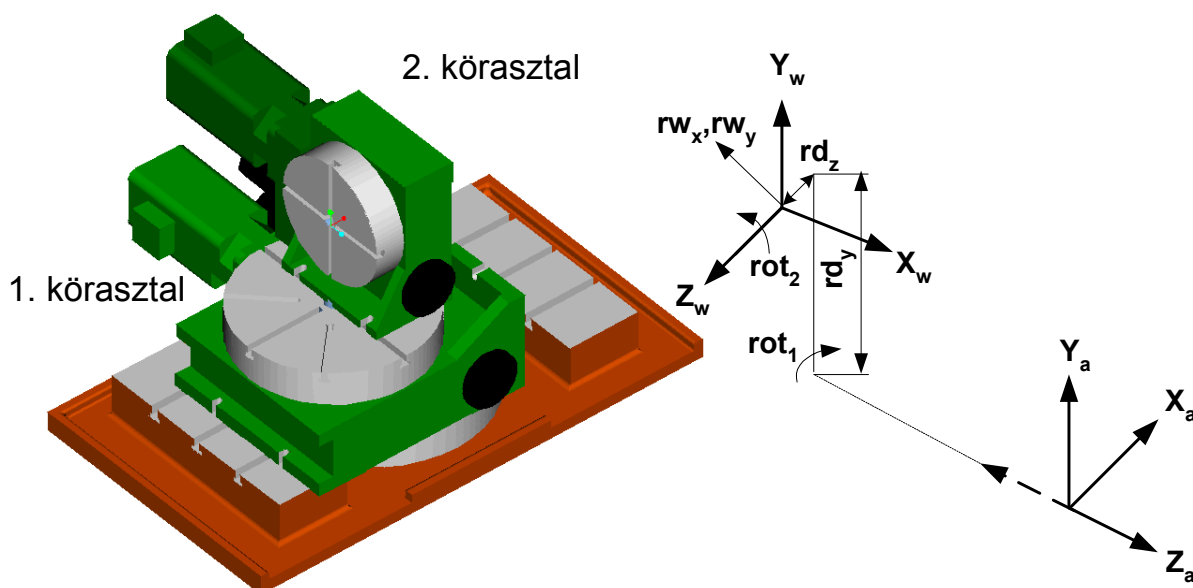
Az 5-tengelyes megmunkálás szimulációjához és ellenőrzéséhez a következő részfeladatokat kell megoldani [27,111]:

- transzformáció,
- linearizáció,
- szerszámgép konfiguráció modellezése és megjelenítése,
- ütközés felügyelet.

2.8.4.1 Transzformáció

A CLData fájlformátum [39], 5-tengelyes szerszámpálya esetén is a munkadarabhoz kötött koordináta-rendszerben értelmezi az egyes szerszámhelyzeteket. Ilyenkor minden egyes szerszámhelyzethez a három koordináta (x, y, z) mellett a szerszámirány is meg van adva (a, b, c)

három koordinátával, $GOTO \setminus x,y,z,a,b,c$ formában. A transzformáció feladata az, hogy az x,y,z,a,b,c 6 koordinátát átalakítsa az adott szerszámgép konfigurációnak megfelelő (x,y,z,rot_1,rot_2) 5 darab koordinátára. Rot_1 és rot_2 jelen esetben a két körasztal elfordulási szögét jelenti, de jelenthet billenési szöget, vagy ritkább esetben főorsó elfordulási szögét. A 16. ábrán látható a Miskolci Egyetem Szerszámgépek Tanszékének műhelycsarnokában üzemelő 2 körasztalos megmunkáló központ méretarányos modellje. Az ábra jobb oldalán az X_a, Y_a, Z_a koordinátarendszer a szerszámgép abszolút koordinátarendszerét, a X_w, Y_w, Z_w pedig a munkadarab koordinátarendszerét reprezentálja (w indexű koordináta rendszer a kettesszámú körasztalhoz kötött). A bal oldali méretarányos modell a Pro/Engineer szoftver segítségével készült.



16. ábra: két körasztallal megvalósított 5-tengelyes gépkonfiguráció modellje látható a baloldalon. A jobb oldalon az 'a' indexű koordinátarendszer a szerszámgép abszolút koordinátarendszerét, a 'w' indexű pedig a második körasztalhoz kötött koordinátarendszert reprezentálja, amelyre a munkadarabot is rögzítik.

A koordináta-transzformáció feladata az, hogy egy adott transzformációs mátrix segítségével, a w -indexű koordinátarendszerből a -indexű koordinátarendszerbe transzformálja az egyes szerszámhelyzeteket. A $T_{w \rightarrow a}$ transzformációs mátrix a következő formában írható fel (a 16. ábra alapján 5 darab 4×4 -es mátrix szorzatából a Maple matematikai szoftver segítségével):

$$\mathbf{T}_{w \rightarrow a} = \begin{bmatrix} -\sin(rot_1) \cos(rot_2) & \sin(rot_1) \sin(rot_2) & \cos(rot_1) & -\sin(rot_1) \cos(rot_2)rw_x + \sin(rot_1) \sin(rot_2)rw_y + \cos(rot_1)rd_z \\ \sin(rot_2) & \cos(rot_2) & 0 & \sin(rot_2)rw_x + \cos(rot_2)rw_y + rd_y \\ \cos(rot_1) \cos(rot_2) & -\cos(rot_1) \sin(rot_2) & \sin(rot_1) & \cos(rot_1) \cos(rot_2)rw_x - \cos(rot_1) \sin(rot_2)rw_y + \sin(rot_1)rd_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

ahol:

- rw_x, rw_y - munkadarab referenciapontjának két koordinátája, a kettesszámú körasztal koordináta-rendszerében.
- rd_y - a kettes körasztal forgástengelye és az egyes körasztal lapja közötti távolság.
- rd_z - az egyes és kettes körasztal Z_a irányú elcsúsztatásából adódó eltolás.

Ha a $\mathbf{T}_{w \rightarrow a}$ mátrixot beszorozzuk a CLDatából nyert (x_w, y_w, z_w, l) koordinátákkal, akkor megkapjuk a szerszámgép koordináta-rendszerébe transzformált pontokat. Egy probléma azonban adódik, mégpedig az, hogy a rot_1 és rot_2 elfordulásokat még nem ismerjük. A két elfordulási szög meghatározása a következő módszerrel lehetséges. Vegyünk fel 'w' koordináta-rendszerben két tetszőleges pontot $(p1_w, p2_w)$, amelyeknek a távolsága 1 egység, ez reprezentálja a szerszám aktuális helyzetét w-ben. Ha ezt a két pontot átranzformáljuk 'a' koordináta-rendszerbe, akkor a kapott két $(p1_a, p2_a)$ pontok különbségének $(0,0,1)$ egységvektornak kell lennie, mivel a szerszám az adott szerszámgép konfiguráció miatt mindig Z_a irányú. Írjuk le ezt a szituációt egy egyenletrendszer segítségével:

$$\mathbf{T}_{w \rightarrow a} \begin{bmatrix} p2_x \\ p2_y \\ p2_z \end{bmatrix}_w - \mathbf{T}_{w \rightarrow a} \begin{bmatrix} p1_x \\ p1_y \\ p1_z \end{bmatrix}_w = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad (9)$$

a következő három egyenletet kapjuk (átalakítások után):

$$\begin{aligned} -\sin(rot_1) \cos(rot_2)(p2_x - p1_x) + \sin(rot_1) \sin(rot_2)(p2_y - p1_y) + \cos(rot_1)(p2_z - p1_z) &= 0, \\ \sin(rot_2)(p1_x - p2_x) + \cos(rot_2)(p1_y - p2_y) &= 0, \\ \cos(rot_1) \cos(rot_2)(p2_x - p1_x) - \cos(rot_1) \sin(rot_2)(p2_y - p1_y) + \sin(rot_1)(p2_z - p1_z) &= 1. \end{aligned} \quad (10)$$

Rot_2 és rot_1 a második és az első egyenletből kifejezhető:

$$rot_2 = -\arctan\left(\frac{p1_y - p2_y}{p1_x - p2_x}\right) = \arctan\left(\frac{b}{a}\right), \quad (11)$$

$$rot_1 = -\arctan\left(\frac{-p2_z + p1_z}{\cos(rot_2)(p1_x - p2_x) + \sin(rot_2)(p2_y - p1_y)}\right) = -\arctan\left(\frac{-c}{\sin(rot_2)b - \cos(rot_2)a}\right). \quad (12)$$

A szerszám irányát kijelölő három koordináta, a CLData a, b, c koordinátái segítségével az egyenletekben szereplő $(p2_* - p1_*)$ tagok kifejezhetők a következőképpen:

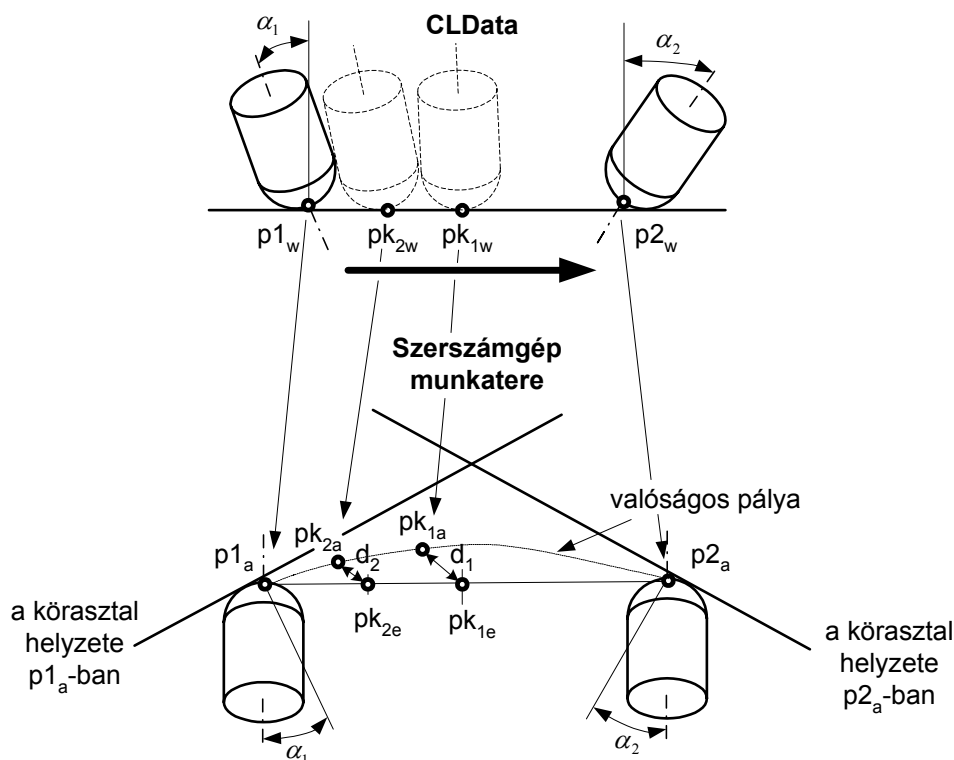
$$p2_x - p1_x = a, p2_y - p1_y = b, p2_z - p1_z = c. \quad (13)$$

A transzformáció folyamata összefoglalva a következő két lépésből áll: 1.) A CLData bemenő x, y, z, a, b, c koordinátái alapján először kiszámítjuk a rot_1 és rot_2 elfordulási szögeket, majd a transzformáció elvégezhető a (8) transzformációs mátrix segítségével. A képletek és az elv helyességét a működő szoftver is igazolja.

2.8.4.2 Linearizáció

A legtöbb 4 és 5 tengelyes szerszámgépen a két forgástengely nem esik egybe a szerszám fizikai végével, csakúgy, mint a jelen példánkban (lásd 16. ábra). Ez azért van így, mert a körasztalok alkalmazása olcsóbb megoldás, mint a főorsó elforgathatóságának megvalósítása. A főorsó billentése merevségi okokból sem célszerű megoldás. Kör és billenthető asztalok alkalmazásával a legtöbb három tengelyes gép öttengelyessé alakítható, ha a vezérlés is képes az 5 tengely egyidejű vezérlésére. A transzformációs eljárás eredményeképpen kapott egyenes vonalú és forgó szánmozgásokat a vezérlés egy NC mondatban kapja meg, a lineáris és forgó mozgások tehát egyenlő arányban és egyenlő időnövekménnyel jelennek meg, és ez azt eredményezi, hogy a szerszám fizikai végében az elmozdulás az alkatrészhez képest hibás görbe pályán történik [27,111].

A linearizáció elvi folyamatát, és szükségességét a 17. ábra alapján lehet nyomon követni. A linearizáció során, a soron következő szerszámpozícióban, köztes pontokban megállapítom a szerszám munkatérbeli pozícióját. Először a felezőpontokban, majd egyre közelebb a kiinduló ponthoz. Ha nagy eltérést tapasztalok, akkor tovább felezem az intervallumot a kiindulóponthoz közeledve, amíg az eltérés (d_i) egy előre megadott értéknél kisebb nem lesz ($d_i \leq \varepsilon$). Amennyiben ez a határfeltétel teljesül, akkor az egész intervallumot felosztom i részre, és ezeket a köztes pontokat felhasználva, rajtuk keresztül vezetem végig a szerszámot $p2_a$ -pontba. A linearizáció során tehát plusz NC sorok keletkeznek, érdekességgként megemlítem, hogy a gyakorlati tapasztalat alapján esetenként 30-40 plusz sort is be kell illeszteni a szerszámpálya két soron következő szerszámpozíciója közé, biztosítva ezzel a megfelelő pontosságot. Az itt bemutatott módszer egy iteratív algoritmusként implementálható, amely kilépési feltétele természetesen: $d_i \leq \varepsilon$.



17. ábra: az ábra a linearizáció folyamatát mutatja. A szerszám programozott pontja a $p1_w$ -ben van. A soron következő CLData sor $p2_w$ -t tartalmazza. Linearizáció alkalmazása nélkül a szaggatott vonallal rajzolt valóságos pálya jönne létre, ami semmiképpen nem felelne meg az előírt pályának.

2.8.4.3 Szerszám gép konfiguráció modellezése és megjelenítése

Az előzőekben már említettem, hogy a szerszám gép geometriai modellezése a már bemutatott OpenGL rendszer segítségével célszerű. Mivel az OpenGL semmilyen geometriai tervezőfunkciót nem támogat, ezért szükséges volt keresni és megtanulni egy olyan CAD rendszert (Pro/Engineer), amellyel rövid idő alatt méretarányosan modellezhető a két körasztalos megmunkáló központ. A módszer általánosítása érdekében kifejlesztettem egy konvertáló osztályt, amely a Pro/Engineer rendszerben megtervezett modelleket képes beolvasni, és megjeleníteni a szimulátorban. A konvertáló osztály pusztán a geometriát és a megjelenítéshez szükséges szín információkat dolgozza fel, az alaksajátosságok felismerését nem támogatja, de segítségével a forgó és állórészeket egymástól függetlenül lehet transzformálni. A Pro/Engineer rendszer képes a benne tervezett alkatrészeket egy speciális SLP formátumban fájlba exportálni. A SLP fájlformátum szövegesen vagy bináris formában tartalmazza a rendereléshez szükséges háromszöghalmaz geometriai információit és az egyes háromszögek színekódjait.

2.8.4.4 Ütközés felügyelet, ütközés érzékelés

Az ütközés felügyelet alatt 5 tengelyes megmunkálás esetén a szerszámgép körasztalainak, a szerszámnak, szerszámbefogónak, munkadarab befogó készüléknek, főorsónak egymással való lehetséges ütközéseinek felügyeletét értem. A körasztalok teljes körbefordulása általában tiltott, amit könnyű ellenőrizni egy egyszerű feltétel beiktatásával, de gyakran előfordulhat, hogy a szerszám belemar a munkadarab felfogó készülékébe, vagy a körasztal forgórészébe, ami komolyan megrongálhatja a gépet és szimuláció nélkül nagyon nehéz előre jelezni. Olyan szoftver eszközt szerettem volna használni, amely képes a háromszögekkel közelített, tetszőleges geometriai formák ütközésének, állítható pontosságú, gyors végrehajtására. A *Solid 2.0* elnevezésű ütközésvizsgáló programkönyvtár alkalmazása teljes mértékben kielégíti a felmerült követelményeket. A rendszer egy úgynevezett *AABB* fa módszerrel dolgozik [38], amely az angol (axis-aligned bounding box) rövidítése. Ez a rendszer a bemenő modellekből egy fastruktúrát képez, és minden modellhez a saját lokális koordináta rendszerének tengelyeihez csatolt, azonos orientációjú határoló téglalapokat definiál. Az ütközések felügyelete előre beállítható pontossággal történik. A *Solid 2.0* alkalmazása abból a szempontból is előnyös, hogy a geometriát és a modellek térbeli helyzetét leíró transzformációs függvényei nagyon hasonlóak az OpenGL rendszerben megszokottakkal.

3 Számítógéppel segített folyamattervezés (CAPP) szabadformájú felületek esetén

A folyamattervezés legfontosabb feladata: műveletelemek sorozatának létrehozása, amely hatékony és megbízható NC program-kódhoz vezet. Ennek elérése a következő három lépésen keresztül történik:

1. Folyamattervezés: alaksajátosság felismerés, folyamat előtervezés.
2. Szerszámpálya generálás: szerszámpálya tervezés, CLData létrehozás.
3. Ellenőrzés: NC geometriai szimuláció és ellenőrzés.

A folyamattervezéskor felhasznált adatok, ismeretek, modellezési és elemzési eljárások határozzák meg, hogy a folyamattervezést *variáns*, *generatív*, *vario-generatív* elvűnek, *intelligensnek*, *automatikusnak* és/vagy *interaktív*nek nevezhetjük. Fontos megjegyezni, hogy tisztán egyetlen megközelítési módot alkalmazni egy rendszer kifejlesztése, alkalmazása során nem érdemes. A probléma felbontása után a részfeladat típusa szerint érdemes eldönteni, hogy annak megoldásához mely eljárást alkalmazzuk. A kiválasztott megközelítési mód részben a program felépítését is meghatározza. Az egyes megközelítési módok egyben magukban hordozzák az információ elosztásának módját is a programlogika, az adatbázis és a tudásbázis között [26,28,15].

A *variáns* módszert akkor alkalmazzák: ha a tudás kész megoldási sémákban, mintákban, ábrázolható, és azok lényegi változtatás nélkül alkalmazhatók a konkrét feladatok megoldására. A *generatív* megközelítés szerint a tudás jól kezelhető, megfelelő környezetfüggetlen és egzakt matematikai modellekben, környezetfüggetlen, és egzakt megoldási módszerekben reprezentálható. E csoportba tartoznak a nagyméretű, de jól dekomponálható modellek is, sőt azok az esetek is, amikor nincsenek egzakt modelljeink, de a feladat megoldására kielégítő heurisztikus algoritmusok alkalmazhatók. A *vario-generatív* módszer lényege, a variáns és a generatív módszer előnyeinek ötvözése, hátrányaik egyidejű kiküszöbölése mellett. Az *intelligens* rendszerek alkalmazásának igénye onnan ered, hogy a technológiai tervezés számos olyan problémát vet fel, amely nehezen algoritmizálható, így az eddig bemutatott módszerekkel igen nagy nehézségek árán, vagy egyáltalán nem oldható meg [15].

A disszertáció nem foglalkozik a szerszámpálya generálás témakörével, az első fejezet pedig már részletesen foglalkozott az ellenőrzés lehetőségeivel.

3.1 Folyamattervezési modellek

A szakirodalomban számos folyamattervezési modell, módszer létezik. Ebben a részben két fontos folyamattervezési modell kerül bemutatásra, főként azért, mert ezek tartoznak szorosan az NC technológia témaköréhez. Ezek a modellek nem csak CAPP rendszerekben alkalmazhatóak hatékonyan, hanem általános tervezési tevékenységek esetén is [26].

3.1.1 Szekvenciális folyamattervezési modell

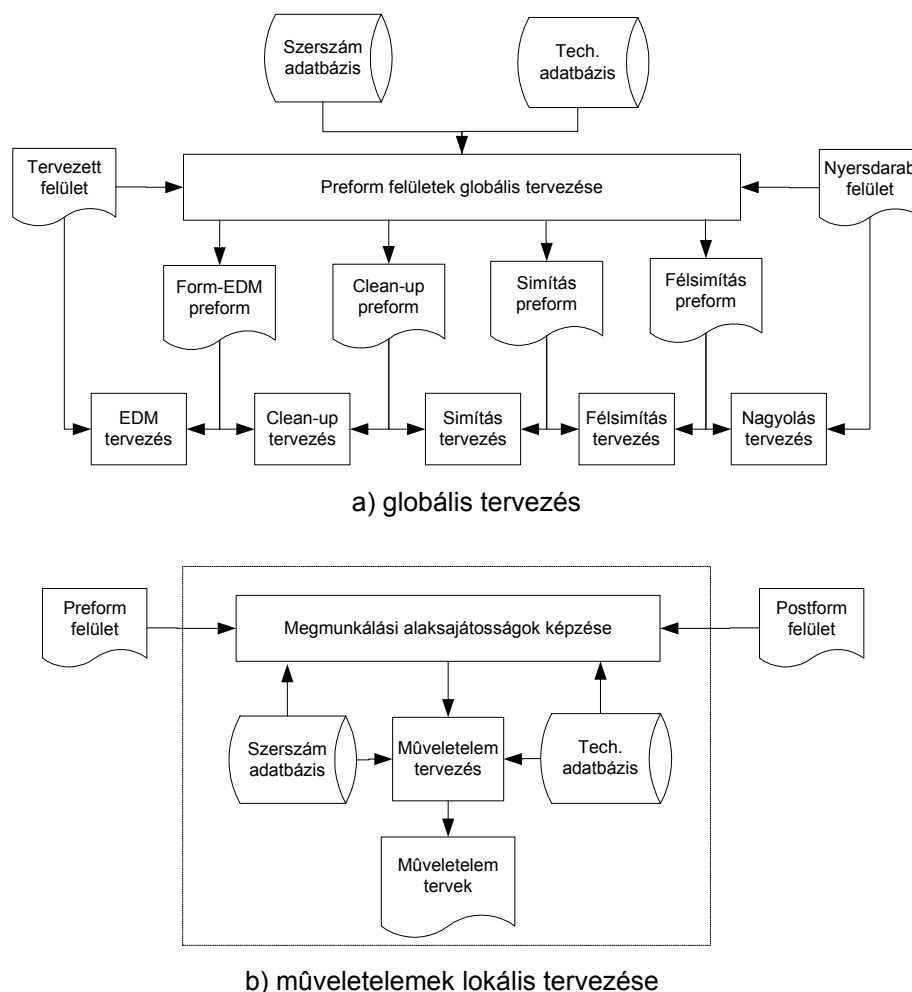
A műveletelemek, műveletekbe csoportosíthatóak. Minden művelet n darab műveletelemből áll. Egy alkatrész folyamatterve általában három műveletet tartalmaz: nagyolás, félsimítás, simítás (de léteznek 5 műveletből álló modern eljárások is.) A szekvenciális folyamattervezési modell a 'rekurzív CAPP logikára' épül. A *MasterCAM* rendszer is ezt a modellt alkalmazta régebbi verzióiban. Ebben a modellben a művelettervezés fordított sorrendben megy végbe. Először a simítás műveletet lehet létrehozni a tervezett felület alapján, majd a félsimítást, a simítás alapján és legvégül a nagyolás művelete generálható a félsimítás alapján.

3.1.2 Hierarchikus folyamattervezési modell

A szekvenciális modell hiányosságait kiküszöbölő eljárás a folyamattervezés ismert hierarchikus felépítésén alapul. A modell megértéséhez a következő két fogalom bevezetése szükséges. *Preform* állapotnak nevezzük a munkadarab 'input' állapotát az egyes műveletelemek, műveletek megkezdése előtt. *Postform* állapotnak nevezzük a munkadarab 'output' állapotát egy adott műveletelem vagy művelet befejezésekor.

A kétszintű hierarchikus folyamattervezési modell a 18. ábrán látható [26]. Az egyes műveletek *preform* állapotait a globális tervezési fázis, az egyes műveletelemeket a lokális tervezési fázis határozza meg.

A globális tervezési modul a következő három komponensből áll: 1) bemenő geometriai fájlok: a tervezett felület és a nyersdarab geometria, 2) adatbázisok: szerszám adatbázis és technológiai adatbázis, 3) globális tervező motor: tervezési szabályok a műveletek definiálásához. Az 18-b. ábra az 5 lokális tervező modul belső felépítését ábrázolja. A lokális tervező modul a következő három komponensből áll: 1) bemenő geometriai fájlok: a *postform* és *preform* felületek, 2) adatbázisok: szerszám adatbázis és technológiai adatbázis, 3) lokális tervező motor: alaksajátosságok megmunkálásának tervezési szabályai.



18. ábra: a hierarchikus folyamattervezési modell, globális és lokális tervezési szintjei [26].

A lokális tervező modul kimenete az adott művelethez tartozó műveletelemek sorozata. A lokális tervező motor generatív CAPP logikát alkalmaz a következő négy lépésben: 1) a bemeneti geometria alaksajátosságokra bontása, 2) az alaksajátosságok konvertálása orientált alaksajátosságokra, 3) a megmunkálási alaksajátosságok képzése az orientált alaksajátosságokból, 4) műveletelemek generálása a megmunkálási alaksajátosságok alapján.

3.2 A technológiai paraméterek szerepe a folyamattervezésben

Az optimális technológiai paraméterek meghatározása a technológiai folyamatok tervezésének egyik klasszikus, rendszeresen visszatérő feladata. Az olyan forgácsoló műveletekre, amelyeknél a leválasztott pillanatnyi forgácskeresztmetszet első közelítésben konstans, az optimalizálási modell hosszú idő óta ismert és számos megoldási módszert is publikáltak [4,45,47,50]. Ilyen statikus modellek felállításával és optimalizálási kérdéseiben

számos magyar kutató is jelentős eredményeket ért el: Horváth Mátyás, Somló János [48,49], Tóth Tibor, Vadász Dénes [44]. Dinamikus optimalizációs eljárásokkal a disszertáció nem foglalkozik, de a teljesség kedvéért megemlítem, hogy már a hetvenes évek elején komoly eredményeket értek el a német Essel, Götz, König, és Jacobs [4,46]

A forgásteget jellegű alkatrészek teljes technológiai folyamatának automatizált tervezésére kifejlesztett TAUPROG-T rendszer a hazai technológiai tervezésemélet sajátos iskolájának kialakulásában fontos szerepet játszott. Rendszertervét *Tóth Tibor, Vadász Dénes, Perger Géza, Detzky Iván* dolgozták ki. A rendszer támogatta a generatív sorrendszintézist hatékony algoritmusokkal, automatikus felfogásmód-tervezést, ráhagyások és műveletközi méretek meghatározását, optimális ráhagyásleválasztási (fogásfelosztási) terv készítését, gépi főidők számítását, technológiai dokumentációk létrehozását [8].

Általánosságban elmondható, hogy a 18-b. ábra lokális műveletelem tervező modulja műveletelemenként külön-külön számított lokális optimumokat határoz meg. Az így meghatározott technológiai paramétereket az NC programba posztprocesszálják. Ez az előidejű tervezés nem számolhat a termelési folyamat olyan valószerű eseményeivel, amelyek a termelés tényleges megvalósítása során merülnek fel. Különösen marás esetén az egyes műveletelemek bonyolult alakú ráhagyási alakzatainak eltávolítása sokszor dinamikus optimalizációs feladatként értelmezhető.

A Miskolci Egyetem Alkalmazott Informatikai Tanszékén működő kutatócsoport sokéves munkája során arra a következtetésre jutott, hogy a technológiai tervezés, a termelési tervezés, műhelyszintű termelésirányítás integrálását, a célszerűen megválasztott intenzitás állapotváltozó jelentősen támogatja [15,19,51,52,53].

Az új módszer előnyei:

1. Az intenzitás változó optimalizálása egyszerűbb feladat, mint a sokváltozós marási vagy esztergálási optimalizációs feladat.
2. Csoporttechnológia alkalmazása esetén a vezéralkatrészre számított optimális intenzitás az egész csoportra érvényes.
3. A CAM alkalmazás esetén a leválasztandó anyagtérfogat (V_m [cm³]) könnyen és gyorsan számítható.

A következő fejezet az új módszer homlokmarásra való alkalmazásával foglalkozik. Az optimális intenzitás kiszámítása esetén egy megmunkálás-szimulátor képes lehet a forgácsolási paraméterek (az előtolás) változtatásával a költséget, termelékenységet az előírt szinten tartani, és az NC kódba posztprocesszálni.

4 Nagyoló homlokmarás költségoptimalizálása anyagleválasztási intenzitás segítségével

A forgácsolási folyamat műveletelemeinek tervezésekor az optimális technológiai paramétereinek meghatározása a gazdaságosság és termelékenység szempontjából kulcsfontosságú feladata. A technológiai paraméterek optimalizálására alkalmazható matematikai modellnek három összetevője van:

- (1) Az adott feladat mértékadó technológiai korlátainak meghatározása;
- (2) Az optimumkritériumként használt célfüggvény meghatározása;
- (3) A szerszáméltartam összefüggés, amely a célfüggvény feltételi egyenlete;

A modell független változói, amelyek egyben az optimalizálandó forgácsolási paraméterek, homlokmarás esetén a fogásmélység (d), a fogásszélesség, és az előtolás (f) vagy fogankénti előtolás (f_z). A nagyoló esztergálás esetére részletes kidolgozott modellt a tanszék kutatócsoportja számos alkalommal publikálta [51,52,53]. A disszertáció keretében vizsgálom meg a homlokmarás esetén felmerülő különbségeket, hasonlóságokat.

4.1 Költségfüggvény forgácsoló megmunkálások esetén

Egy forgácsolási művelet elem költsége az anyagleválasztási és a szerszám-elhasználódási költségek összege [14]:

$$K_{\Sigma} = K_m + K_s \quad [\text{Ft}] \quad (14)$$

ahol K_m a megmunkálás költsége:

$$K_m = k t_g \quad [\text{Ft}] \quad (15)$$

ahol:

k = a környezeti percköltség [Ft/min]

t_g = megmunkálási idő [min]

$$t_g = \frac{V_m}{Q} \quad [\text{min}] \quad (16)$$

A (16)-ban a V_m leválasztandó térfogat (működési térfogat) homlokmarás esetén a szerszám és a munkadarab elhelyezésétől (d, w) és a forgácsolási úthosszától (l_m) függ.

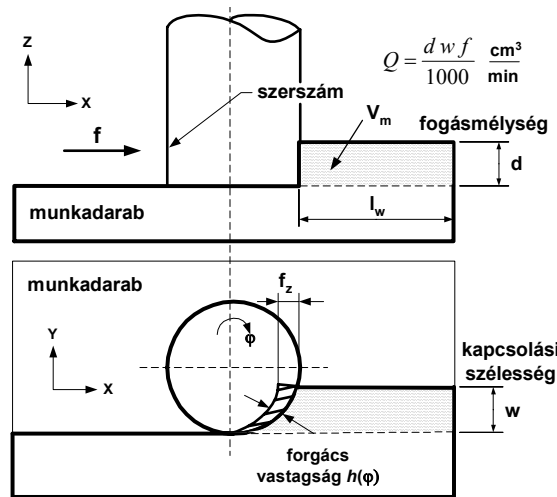
$$V_m = \frac{d w l_m}{1000} \quad [\text{cm}^3] \quad (17)$$

A (16)-ban szereplő Q anyagleválasztási intenzitás definíciója természetesen eltér az esztergáláshoz bevezetettől, amely homlokmarás esetén a fogásmélység, a kapcsolási méret és az előtolási sebesség szorzata: [54,55]

$$Q = \frac{d w f}{1000} \quad [\text{cm}^3/\text{min}] \quad (18)$$

A szerszám elhasználódás költsége ugyanolyan alakú képlettel számítható, mint esztergálás esetén:

$$K_s = K_{sz} N_{cs} \quad [\text{Ft}] \quad (19)$$



19. ábra: homlokmarás a jellemző technológiai jellemzőivel (oldal és felülnézetben)

K_{sz} a szerszámköltség:

$$K_{sz} = K_{sb} + k t_{cs} \quad [\text{Ft}] \quad (20)$$

ahol:

K_{sb} = a szerszám alkalmazásával kapcsolatos teljes költség (beszerzés, tárolás, hulladék)
[Ft]

t_{cs} = a szerszám cseréjéhez szükséges átlagos idő [min].

Az élek cseréjének száma a működési (gépi) idő alatt:

$$N_{cs} = \frac{t_g}{T}. \quad (21)$$

A szerszáméltartam számítása a [1,4] szerinti bővített Taylor éltartam egyenlet alapján, célszerűen reciprok alakban írva:

$$\frac{1}{T} = \left[\frac{d^{x_v} f_z^{y_v} w^{z_v} z^{p_v} v}{C_v D_c^{w_v}} \right]^{\frac{1}{m}}. \quad (22)$$

A következő lépésben az intenzitás képletét alakítsuk át úgy, hogy a Taylor egyenlet fogankénti előtolás és forgácsolási sebesség paraméterei is szerepeljenek benne:

$$Q = \frac{d w f}{1000} = \frac{d w f_z z n}{1000} = \frac{d w f_z z}{1000} \frac{1000 v}{\pi \cdot D_c} = d w f_z z \frac{v}{\pi D_c} \quad [\text{cm}^3/\text{min}] \quad (23)$$

Fejezzük ki (22)-et az anyagleválasztási intenzitás segítségével

$$\frac{1}{T} = \left[\frac{d^{x_v} f_z^{y_v} w^{z_v} z^{p_v} v}{C_v D_c^{w_v}} \right]^{\frac{1}{m}} = \left[\frac{Q \pi}{C_v d^{1-x_v} f_z^{1-y_v} w^{1-z_v} z^{1-p_v} D_c^{w_v-1}} \right]^{\frac{1}{m}} \quad (24)$$

Az előbbiek szerint értelmezett teljes költség kifejezhető a forgácsleválasztás intenzitásával és a többi paraméterrel:

$$K_{\Sigma} = K_{\Sigma}(Q, d, f_z, w, D_c, z, V_m, k, K_{sb}, N_{\text{él}}, t_{cs}, C_v, m, x_v, y_v, z_v, p_v, w_v)$$

$$K_{\Sigma} = k \frac{V_m}{Q} + (K_{sb} + k t_{cs}) \frac{V_m}{Q} \frac{Q^{\frac{1}{m}}}{\left[C_v \pi^{-1} d^{1-x_v} w^{1-z_v} z^{1-p_v} D_c^{w_v-1} \right]^{\frac{1}{m}} f_z^{\frac{1-y_v}{m}}}. \quad (25)$$

4.2 Fajlagos költségekvalens idő

A költségfüggvényt egyidejűleg vonatkoztathatjuk a környezeti percköltségre és a működési térfogatra, ezáltal a fajlagos költségekvalens időt kapjuk:

$$\tau(Q, f_z) = \frac{K_\Sigma}{kV_m} = \frac{1}{Q} + \left(\frac{K_{sb}}{k} + t_{cs} \right) \frac{Q^{\frac{1}{m}-1}}{\left[C_v \pi^{-1} d^{1-x_v} w^{1-z_v} z^{1-p_v} D_c^{w_v-1} \right]^{\frac{1}{m}} f_z^{\frac{1-y_v}{m}}} \text{ [min/cm}^3\text{]} \quad (26)$$

A $\tau(Q, f_z)$ függvényt azért célszerű bevezetni, mert egy adott tartományon vett minimuma közvetlenül megadja a minimális költséghez tartozó fajlagos anyagleválasztási időt, vagyis egy cm^3 anyag leválasztásához szükséges időt, tehát egyszerre fejez ki gazdasági és technológiai információt.

Vezessük be a következő jelölést:

$$C_s^{\frac{1}{m}} = \frac{\left[C_v \pi^{-1} d^{1-x_v} w^{1-z_v} z^{1-p_v} D_c^{w_v-1} \right]^{\frac{1}{m}}}{\frac{K_{sb}}{k} + t_{cs}} \quad (27)$$

amiből:

$$C_s = \frac{C_v d^{1-x_v} w^{1-z_v} z^{1-p_v} D_c^{w_v-1}}{\pi \left(\frac{K_{sb}}{k} + t_{cs} \right)^m}. \quad (28)$$

A C_s paraméter a (26) összefüggés zárójeles paramétereit foglalja össze egy olyan új tényezőbe, amely az adott szerszám komplex jellemzője. Ezek a paraméterek egy műveletelemen belül nem változnak. Helyettesítsük a C_s tényezőt a (25)-be:

$$\tau(Q, f_z) = \frac{1}{Q} + \frac{Q^{\frac{1}{m}-1}}{C_s^{\frac{1}{m}} f_z^{\frac{1-y_v}{m}}} = \frac{1}{Q} + \frac{Q^{\frac{1}{m}-1}}{\left(C_s f_z^{1-y_v} \right)^{\frac{1}{m}}}. \quad (29)$$

Vezessük be a következő, szintén térfogat intenzitás dimenziójú változót:

$$R = C_s f_z^{1-y_v}. \quad [\text{cm}^3/\text{min}] \quad (30)$$

Ezt behelyettesítve a (29)-be:

$$\tau(Q, R) = \frac{1}{Q} + \frac{Q^{\frac{1}{m}-1}}{R^{\frac{1}{m}}}. \quad [\text{min/cm}^3] \quad (31)$$

Alkalmazva az $\frac{1}{m} = q$ jelölést, a (31) végül a következő egyszerű alakba megy át:

$$\tau(Q, R) = \frac{1}{Q} + \frac{Q^{q-1}}{R^q}. \quad [\text{min/cm}^3] \quad (32)$$

(32) formailag teljesen azonos a [14]-ben levezetettel. Eltéréseket az anyagleválasztási intenzitás (Q) definiálásában valamint a szerszáméltartam-egyenlet és a szerszám komplex jellemzőjében (C_s) találhatunk. Ezért a további levezetésektől eltekintek.

A [14]-ben levezetett végeredmény a következő:

$$u = Q\tau - 1, \quad (33)$$

bevezetése után a megoldandó egyenlet a következő alakú lesz:

$$2 + (q-1) - [3(q-1) + (q-1)^2]u + (q-1)^2 u^2 - q(q+1)u^{\frac{(q+2)}{q}} + q(q-1)u^{\frac{2(q+1)}{q}} = 0. \quad (34)$$

A szintvonalak paraméteres egyenlete ($\tau = \text{const}$):

$$R(Q, \tau) = \frac{Q}{(Q\tau - 1)^m}; \quad (Q\tau > 1). \quad (35)$$

A (34) egyenletet valamilyen közelítő módszerrel u -ra megoldva a (33) összefüggésből $Q\tau$ meghatározható és a (35) alapján bármelyik szintvonalra érvényes $R = R(Q)$ függvénykapcsolat megkapható. Ha nincsenek korlátok, azaz a feladat nem korlátozza a forgácsolási paraméterek értékeit (Q, R) síkon, akkor a völgyvonal jelenti a felületen a legalacsonyabban fekvő pontok halmazát, a feladat megoldását. A gyakorlatban azonban mindig vannak technológiai korlátok, ekkor viszont a fajlagos költség-egyenértékű időfüggvény feltételes minimuma annak a pontnak a környezetében található, ahol a völgyvonal kilép a korlátos tartomány határán. Ennek a heurisztikus szabálynak az alkalmazásához szükséges a völgyvonal egyenlete. [14,19]

4.3 A korlátfüggvények

A következő lépés a forgácsolási paraméterek értékeinek korlátozása a (Q,R) síkon. Először az éltartam korlátot határozzuk meg.

4.3.1 Éltartamkorlát

Nehezen beszerezhető, vagy drága szerszám használata esetén, egy viszonylag hosszabb minimális éltartamértéket is megadhatunk, és arra keressük a választ, hogy milyen (Q,R) esetén tartható ez a viszonylag magas éltartamkorlát.

A bővített Taylor egyenlet alapján a szerszám éltartamkorlát alsó megengedett értéke:

$$T_a^m = \frac{C_v D_c^{w_v}}{d^{x_v} f_z^{y_v} w^{z_v} z^{p_v} v} = \frac{C_v d^{1-x_v} f_z^{1-y_v} w^{1-z_v} z^{1-p_v} D_c^{w_v-1}}{Q \pi}. \quad (36)$$

(28) és (30) felhasználásával azt kapjuk, hogy:

$$R = \frac{C_v d^{1-x_v} w^{1-z_v} z^{1-p_v} D_c^{w_v-1} f_z^{1-y_v}}{\pi (t_s + t_{cs})^m}, \quad (37)$$

ahol $t_s = \frac{K_{sb}}{k}$ a *szerszámköltség-ekvivalens* idő.

Mivel (36) és (37) egyenletek számlálója azonos, ezért:

$$\begin{aligned} R \pi (t_s + t_{cs})^m &= Q T_a^m \pi, \\ R &= \left(\frac{T_a}{(t_s + t_{cs})} \right)^m Q = C_T Q. \end{aligned} \quad (38)$$

Látható hogy az alsó éltartamkorlát a (Q,R) síkon egy, az origón átmenő egyenesnek felel meg. Könnyen belátható, hogy létezik egy T_0 szerszáméltartam bármely adott műveletelemhez, amelynél a szerszáméltartam egyenese éppen az optimumvonal egyenesével esik egybe. Ekkor $C_T = m_1$, vagyis az optimumvonal iránytangense:

$$T_0 = (t_s + t_{cs}) m_1^{\frac{1}{m}}. \quad (39)$$

4.3.2 Teljesítménykorlát

A nyomaték számításánál értelmezhető maximális és átlagos forgácsolási nyomaték, amely a maximális illetve átlagos fogácsoló erő alapján számítható [1]. Teljesítményszámítás esetén az átlagos nyomatékra, illetve forgácsoló erőre van szükség (szilárdsági számítások esetén pedig a maximális nyomaték illetve forgácsoló erő szükséges).

Az átlagos marási teljesítmény számítható [54,1,5] az átlagos főforgácsoló erő és a forgácsolási sebesség szorzata alapján. Egy adott P_a teljesítménykorlát esetén:

$$\begin{aligned} P_a &= F_c v, \quad F_c = k_c d w f_z \frac{z}{D_c \pi}, \\ P_a &= \frac{k_c d w f_z z}{D_c \pi 60 \cdot 10^3} v = K Q \quad [kW]. \end{aligned} \quad (40)$$

Ha adott a szerszámgép teljesítménye P_a , akkor ez az érték $Q = \frac{P_a}{K}$ anyagleválasztási intenzitást tesz lehetővé, amely (Q, R) síkon egy függőleges egyenesként jelenik meg, tehát jobb oldalról határolja a keresési tartományt.

4.3.3 Forgácsolási sebesség korlát

A forgácsolási sebesség egy előre rögzített v_{min} , v_{max} tartományban változhat. Határozzuk meg ezen alsó, illetve felső korlát képét a (Q, R) síkban.

Mivel $Q = d w f = d w f_z z \frac{v}{\pi D_c}$ ezért most fejezzük ki f_z -t és helyettesítsük be $R = C_s f_z^{1-y_v}$ -be (v_{min}, v_{max} behelyettesíthető a v helyére az egyenletben)

$$R = C_s \left(\frac{Q}{d w z \frac{v}{\pi D_c}} \right)^{1-y_v} = \frac{C_s}{\left(d w z \frac{v}{\pi D_c} \right)^{1-y_v}} Q^{1-y_v} = C_v Q^{1-y_v}. \quad (41)$$

Látható tehát (41) alapján, hogy a sebességkorlátok képe hatványfüggvény (Q, R) -ben.

4.3.4 Szerszám megengedhető elhajlása

A szerszám, F_c forgácsolási erő hatására δ mm-t hajlik el. Legyen δ_f a legnagyobb megengedhető szerszámelhajlás:

$$\delta_f = \frac{F_c L^3}{3 E I}, \quad I = \frac{\pi D_c^4}{64}, \quad (42)$$

$$\delta_f = \frac{64}{3 \pi E} \frac{F_c L^3}{D_c^4} = \frac{64}{3 \pi^2} \frac{E L^3 k_c d w f_z z}{D_c^3}. \quad (43)$$

Fejezzük ki f_z -t (30) ből:

$$f_z = \left(\frac{R}{C_s} \right)^{\left(\frac{-1}{y_v - 1} \right)},$$

majd helyettesítsük vissza (43)-be és fejezzük ki R -et:

$$R = C_s \left(\frac{3 \delta_f D_c^3 \pi^2}{64 E L^3 k_c d w z} \right)^{1-y_v} = C_s C_\delta \delta_f^{1-y_v}, \quad (44)$$

amely (Q, R) síkon egy vízszintes egyenesként ábrázolható.

4.4 Tipikus döntési folyamatok a (Q, R, τ) állapotterben

4.4.1 Direkt feladat

A technológiai folyamattervezés ún. direkt feladata három tipikus döntési helyzethez vezet. A direkt szó itt arra utal, hogy a mindenkor feltételrendszert kelégitő optimális anyagleválasztási intenzitást (Q_{opt}) közvetlenül keressük. [14,19]

4.4.1.1 Szabályos keresési tartomány

A korlátfüggvények (Q, R) síkon összefoglalt képeit, a következő 20. ábra szemlélteti. Ezek a függvények jelölik ki a keresési tartományt (a bevonalkázott terület) amely a technológiai

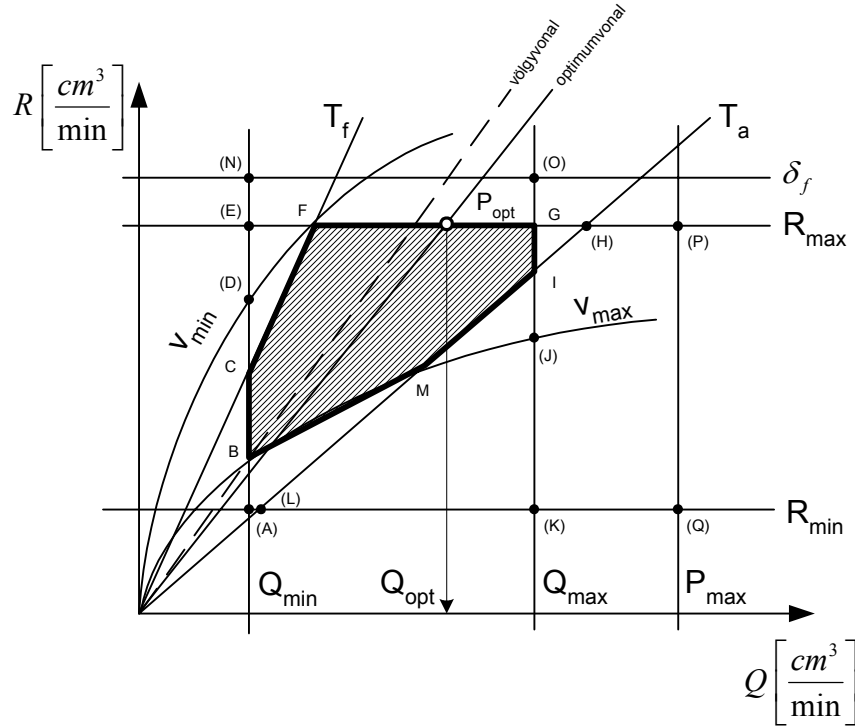
szempontból megengedett Q és R értékek halmaza. Az optimum esélyes pontok mindig a tartomány határán vannak. A szélsőérték meghatározásában fontos szerepe van az optimum és völgyvonalnak.

Az optimumvonal egyenlete az $R = \text{MIN}(\partial \tau / \partial Q)$ feltételből $R = m_2 Q$ egyenesnek adódik:

$$\text{MIN}\left(\frac{\partial \tau}{\partial Q}\right) = \frac{1}{Q^2} + \frac{(q-1)Q^{q-2}}{R^q} = 0, \quad (45)$$

amiből:

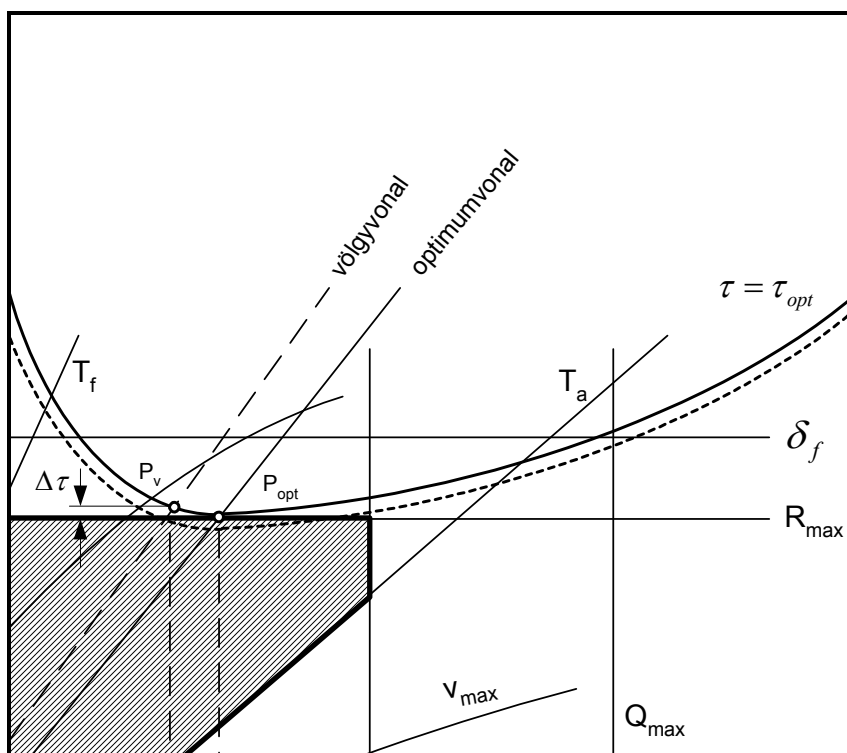
$$R = (q-1)^{\frac{1}{q}} Q = m_2 Q. \quad (46)$$



20. ábra: optimalizációs tartomány a (Q, R, τ) állapottérben, a mértékadó korlátok feltüntetésével ($v_{\min}$, $v_{\max}$, $Q_{\min}$, $Q_{\max}$, $R_{\max}$, $R_{\min}$, $P_{\max}$, δ_f , T_a , T_f). Az ábrán zárójelben írt betűkkel jelöltük azokat a pontokat, amelyek szóba jöhetnek a mindenkor keresési tartomány meghatározásakor, de a jelen esetben nem tartoznak az aktuális optimalizációs tartományhoz.

A völgyvonal és az optimumvonal mentén az origótól távolodva a fajlagos költségekivalens idő $\tau(Q, R)$ függvénye monoton csökken és $R_{\max}$ -nál ütközik a tartomány

határába. Ha a feladat nem tartalmaz korlátozó feltételeket, akkor a völgyvonal határozza meg a felület legalacsonyabban fekvő pontjait, azaz az optimális (Q,R) ponthalmazt. Ha csak Q_{max} lenne a mértékadó korlát és eltekintenénk a többi korláttól, akkor az optimális intenzitás megegyezne a maximális intenzitással $Q_{opt} = Q_{max}$, mivel a völgyvonal és a Q_{max} egyenes metszéspontjához tartozik a legkisebb költség. Mivel a valóságban R_{max} mindig fontos mértékadó korlát és képe vízszintes egyenes, ezért a 21. ábra alapján leolvasható, hogy ebben az esetben nem a völgyvonal, hanem az optimumvonal jelöli ki a minimális költségű intenzitást, mivel minden pontja egy-egy szintvonal vízszintes érintőjéhez tartozik, tehát hozzá mindig alacsonyabb költség tartozik, mint a P_v -hez (lásd 21. ábra).

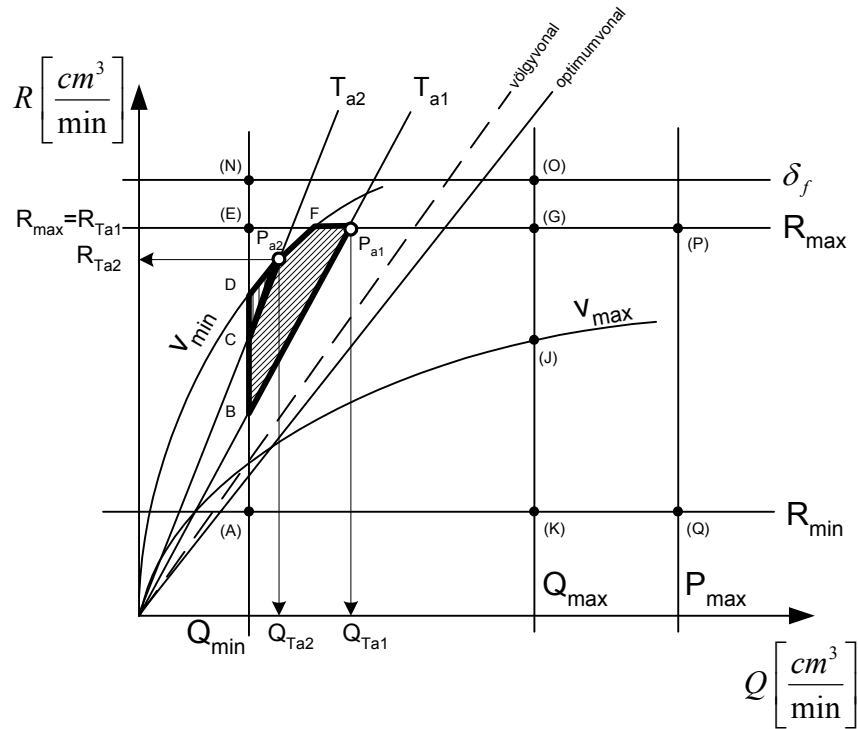


21. ábra: optimalizálási tartomány a (Q,R,τ) állapotterben, kinagyítva az optimumesélyes tartomány közelében. Az ábra alapján könnyen belátható, hogyha létezik vízszintes mértékadó korlát, akkor az optimumvonalhoz tartozó P_{opt} -hoz alacsonyabb költség tartozik, mint a völgyvonal által kijelölt P_v ponthoz. Ezért a P_{opt} pont jelenti az optimalizálási feladat megoldását.

Az optimalizálási tartományt definíciószerűen *szabályosnak* tekintjük, ha a völgyvonal és az optimumvonal egyenesei a tartományon belül haladnak. Az optimumpont ilyenkor az R_{max} egyenes és az optimumvonal metszéspontja.

4.4.1.2 Magas éltartam alsókorlát

Elfajult, degenerált keresési tartomány alakulhat ki, ha a termelésirányítási rendszer valamilyen ok miatt magas éltartam alsó korlátot ír elő, nagy értékű, vagy nehezen beszerezhető szerszám esetén. Az éltartamkorlát képe a keresési síkon C_T meredekségű egyenes (38). Amennyiben ennek meredeksége nagyobb, mint a völgyvonal és az optimumvonal meredeksége, akkor az éltartamkorlát mértékadó korláttá válik. Ilyenkor az optimumvonal által meghatározott optimális intenzitás általában nem érhető el, vagy kedvezőtlenebb esetben R_{max} sem érhető el. A 22. ábra két olyan esetet szemléltet, amikor az éltartamkorlát válik mértékadóvá. T_{a1} esetében R_{max} még elérhető, de T_{a2} esetében már csak egy kisebb R_{Ta2} érték érhető el, ami v_{min} és T_{a2} metszéspontjához tartozik.

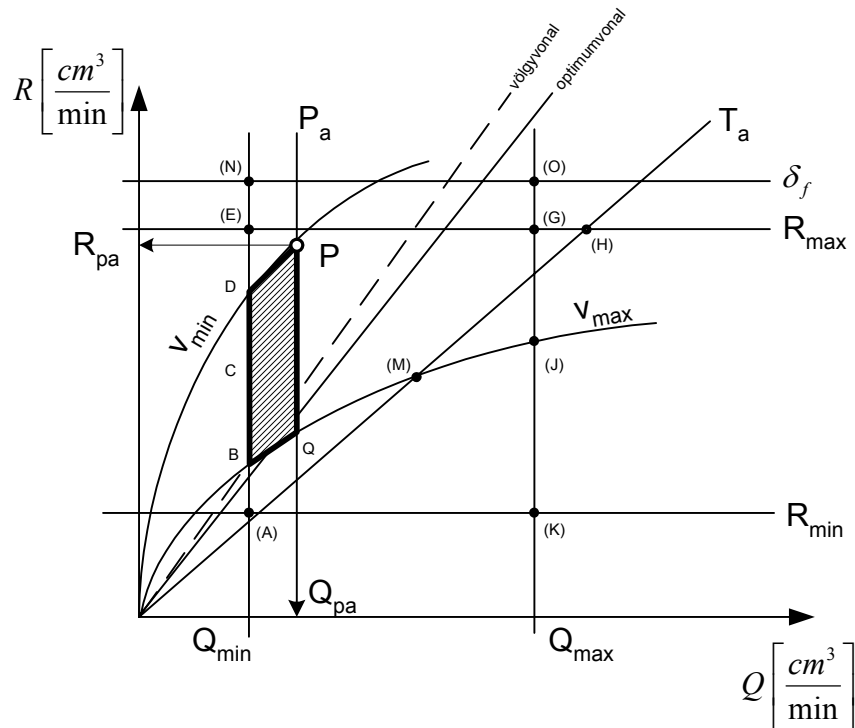


22. ábra: nagy értékű vagy nehezen beszerezhető szerszám esetén a T_a megengedett éltartam viszonylag magas értéke erősen korlátozza az anyagleválasztás intenzitását. Előírt T_{a1} éltartamkorlát esetén $BDFP_{a1}$ és T_{a2} esetén CDP_{a2} tartományok degeneráltak, a völgyvonal és az optimumvonal mindkét esetben kívül esik a keresési tartományon.

A modell alapján kiadódó intenzitást ilyenkor feltételes kvázioptimális megoldásnak nevezzük.

4.4.1.3 Alacsony teljesítménykorlát

A harmadik jellegzetes esetben a kiválasztott szerszámgép teljesítménye nem elegendő, ilyenkor az optimális intenzitás nem érhető el. Alacsony P_a teljesítménykorlát jellegzetes degenerált keresési tartományhoz vezet (23. ábra), ilyenkor a kvázioptimális pont P a forgácsolási sebesség alsó korlátja és a teljesítménykorlát metszéspontjába csúszhat. Nagyobb teljesítmény esetén P felkerülhet az R_{max} vonalára.



23. ábra: ha a szerszámgép teljesítménye jelenti a mértékadó korlátot, akkor a keresési tartomány jellegzetes módon torzul, a kvázioptimális intenzitás az ábra alapján: $P(Q_{pa}, R_{pa})$. Nagyobb teljesítmény esetén P felkerülhet az R_{max} vonalára.

4.4.2 Indirekt feladat

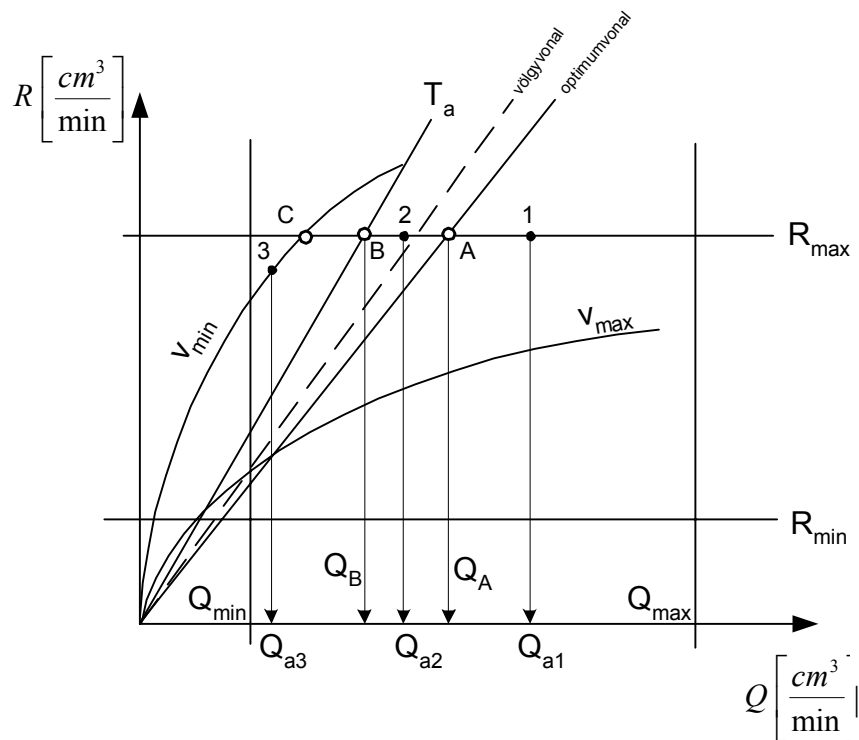
A (Q, R, τ) állapottérben való optimalizálás igazi előnyei vagy az előtervezésben mutatkoznak meg, amikor megmunkálási időket jó közelítéssel számíthatunk részletes technológiai paraméter-értékek ismerete nélkül, vagy egy integrált CAD/CAPP/PPS, ill. CIM rendszer alkalmazásakor. [14,23,24]

Az indirekt feladat akkor kerül előtérbe, amikor "szűk keresztmetszet" keletkezik a rugalmas kissorozatú gyártás cella- vagy műhely szintjén. A feladat célja, ennek az elhárítása,

mivel a "szűk keresztmetszet" határidő túllépést okozhat, aminek komoly következményei lehetnek nemlineáris költségfüggvény (késedelmi kamat) esetén. Vagyis a Q intenzitás növelése a műhely költségeit csökkentheti, annak ellenére, hogy a "szűk keresztmetszet", mint munkahely, növelt költséggel dolgozik.

Az indirekt feladathoz kapcsolódó döntéseket a 24. ábra szemlélteti.

1. A PPS rendszer által előírt Q_{a1} intenzitás beállítását nem korlátozza a feltételrendszer, és Q_{a1} az optimumvonal által kijelölt optimumponttól (A) jobbra helyezkedik el, akkor a költségoptimalis intenzitás Q_{a1} lesz és a hozzá tartozó R érték R_{max} lesz. Ha a PPS által előzetesen előírt intenzitás Q_{a2} és az optimumponttól (A) balra helyezkedik el, akkor előnyösebb $Q_A (> Q_{a2})$ intenzitást alkalmazni, mivel alacsonyabb költség tartozik hozzá.
2. A PPS rendszer által előírt Q_{a2} intenzitás mellett alsó éltartamkorlát T_a is adott, akkor az R_{max} korláton lévő B pont helyzete a döntő. Ha $Q_{a2} > Q_B$ akkor az előírt intenzitás nem tartható. A $Q_{a3} < Q_B$ esetben, célszerű a keresési tartomány határán előbb a C pontba majd onnan B pontba haladni. Könnyen belátható, hogy ebben az esetben a B ponthoz tartozó koordináták (Q_B, R_{max}) adják, a legkedvezőbb fajlagos költségekvivalens időt.



24. ábra: az indirekt feladat néhány tipikus döntési esete. (1) A PPS rendszer által előírt Q_{a1} intenzitás beállítását nem korlátozza a feltételrendszer, és Q_{a1}

az optimumvonal által kijelöl optimumponttól (A) jobbra helyezkedik el, akkor a költségoptimalis intenzitás Q_{a1} lesz és a hozzá tartozó R érték R_{max} lesz. Ha a PPS által előzetesen előírt intenzitás Q_{a2} és az optimumponttól (A) balra helyezkedik el, akkor előnyösebb $Q_A (> Q_{a2})$ intenzitást alkalmazni. (2) Ha Q_{a2} intenzitás mellett alsó éltartamkorlát T_a is adott, akkor az R_{max} korláton lévő B pont helyzete a döntő. (2a) Ha $Q_{a2} > Q_B$ akkor az előírt intenzitás nem tartható. (2b) A $Q_{a3} < Q_B$ esetben, célszerű a keresési tartomány határán előbb a C pontba majd onnan B pontba haladni. Könnyen belátható, hogy ebben az esetben a B ponthoz tartozó koordináták (Q_B, R_{max}) adják, a legkedvezőbb fajlagos költségekvivalens időt.

4.5 A módszer bemutatása egy konkrét példa alapján

Az optimalizációs módszert a következő gyakorlati példán szemléltetjük: a művelet legyen nagyoló homlokmarás, melynek technológiai adatait az [57]-ben közzétett táblázatok tartalmazzák. Legyen a nagyoló szerszám egy 40 mm átmérőjű, $R3$ gyorsacél homlokmaró. A munkadarab anyaga: szénacél $R_m = 735 \text{ Mpa}$.

Érvényességi tartomány:

$$\begin{aligned} f_{z \min} &= 0.2 \text{ mm} & f_{z \max} &= 0.5 \text{ mm} \\ v_{\min} &= 10 \text{ m/min}^{-1} & v_{\max} &= 20 \text{ m/min}^{-1} \end{aligned}$$

Géphasználattal és szerszám-elhasználódással kapcsolatos költségek:

k	$= 166 \text{ Ft/min}$	– géphasználat percköltsége
K_{sb}	$= 5000 \text{ Ft}$	– szerszámbeszerzés költsége
t_{cs}	$= 5 \text{ min}$	– szerszámcsere idő
z	$= 6$	– szerszám marófogainak a száma

Technológiai adatok:

d	$= 6 \text{ mm}$	– fogásmélység
w	$= 20 \text{ mm}$	– fogásszélesség
D_c	$= 40 \text{ mm}$	– szerszám átmérő

Taylor éltartam egyenlet állandói [57] alapján:

$$\begin{aligned} C_v &= 41 \\ w_v &= 0.45 \end{aligned}$$

$$\begin{aligned}
m &= 0.2 \\
y_v &= 0.4 \\
x_v &= 0.1 \\
z_v &= 0.15 \\
p_v &= 0.1
\end{aligned}$$

A megoldás első lépése a C_s kiszámítása (28) alapján:

$$C_s = \frac{C_v d^{1-x_v} w^{1-z_v} z^{1-p_v} D_c^{w_v-1}}{\pi \left(\frac{K_{sb}}{k} + t_{cs} \right)^m} \approx 270.356. \quad (47)$$

Továbbá: $R = C_s f_z^{1-y_v} = 270.356 f_z^{0.6}$ és $q = \frac{1}{m} = 5$. Így (32) a következőképpen írható fel:

$$\tau(Q, R) = \frac{1}{Q} + \frac{Q^4}{R^5}. \quad (48)$$

Az $u = Q\tau - 1$ jelölést alkalmazva $q = 5$ -re aktualizáljuk a (34) egyenletet:

$$40u^{2.4} + 4u^2 - 10u^{1.4} - 56u + 12 = 0 \quad (49)$$

Az egyenletet megoldva két zérushely adódik: $u_1 \approx 0.215$ és $u_2 = 1.206$, amelyek alapján:

$$R_1 = \frac{Q}{(Q\tau - 1)^{0.2}} = \frac{Q}{0.214^{0.2}} = 1.358 Q \quad (50)$$

$$R_2 = \frac{Q}{(Q\tau - 1)^{0.2}} = \frac{Q}{1.206^{0.2}} = 1.038 Q \quad (51)$$

(50) (51) függvények, amelyek a Q, R síkon egyenesek, a (32) célfüggvénybe behelyettesítve a következőket eredményezi:

$$\tau(Q, R_1) = \frac{1}{Q} + \frac{Q^4}{(1.358 Q)^5} = \frac{1}{Q} \left(1 + \frac{1}{1.358^5} \right) = 1.216 \frac{1}{Q}, \quad (52)$$

$$\tau(Q, R_2) = \frac{1}{Q} + \frac{Q^4}{(1.038 Q)^5} = \frac{1}{Q} \left(1 + \frac{1}{1.038^5} \right) = 1.829 \frac{1}{Q}. \quad (53)$$

Mivel $1.216 < 1.829$, ezért $R_1 = 1.358Q$ lesz a völgyvonal egyenlete, a feladat megoldása. Ha a völgyvonal egyenletét helyettesítjük a célfüggvénybe (52), akkor az egyenlet kifejezi adott Q intenzitás melletti legkisebb elérhető *költségekvivalens* időfüggvény értékét, azaz a legkisebb költséget. Optimum feltételként a minimális költség volt megadva, adott technológiai folyamatra jellemző korlátok figyelembevételével.

Ha a feladat tartalmaz korlátfeltételeket, akkor a lekorlátozott Q, R síkon, az optimumvonal határozza meg az optimális megoldást, a minimális költséget, ha a keresési tartomány szabályosnak tekinthető.

A fogankénti előtölés maximális értékéből adódó korlátfüggvény egy vízszintes egyenes (Q, R) síkon, mivel $R_{\max} = C_s f_{z \max}^{1-y_v}$. Tehát, ha a maximális előtölés korlát értelmezett, akkor ezzel R maximális értéke is korlátozott

Előállíthatjuk az:

$$R = \text{MIN} \left(\frac{\partial \tau}{\partial Q} \right) \quad (54)$$

függvényt:

$$\frac{\partial \tau}{\partial Q} = -\frac{1}{Q^2} + \frac{4Q^3}{R^5} = 0; \quad (55)$$

amiből $R = \sqrt[5]{4} Q$ adódik.

Ezt behelyettesítve a (32) célfüggvénybe:

$$\tau(Q, R_1) = \frac{1}{Q} + \frac{Q^4}{(\sqrt[5]{4} Q)^5} = \frac{1}{Q} \left(1 + \frac{1}{4} \right) = 1.25 \frac{1}{Q} \quad (56)$$

(52) és (56) alapján belátható, hogy azonos Q esetén a völgyvonal mentén alacsonyabb költséget kapunk, mint az optimumvonal mentén, de ha létezik $R_{\max}$ korlát akkor mégis az optimumvonal fogja meghatározni a feladat megoldását (21. ábra alapján).

Tehát a völgyvonalhoz tartozó intenzitás:

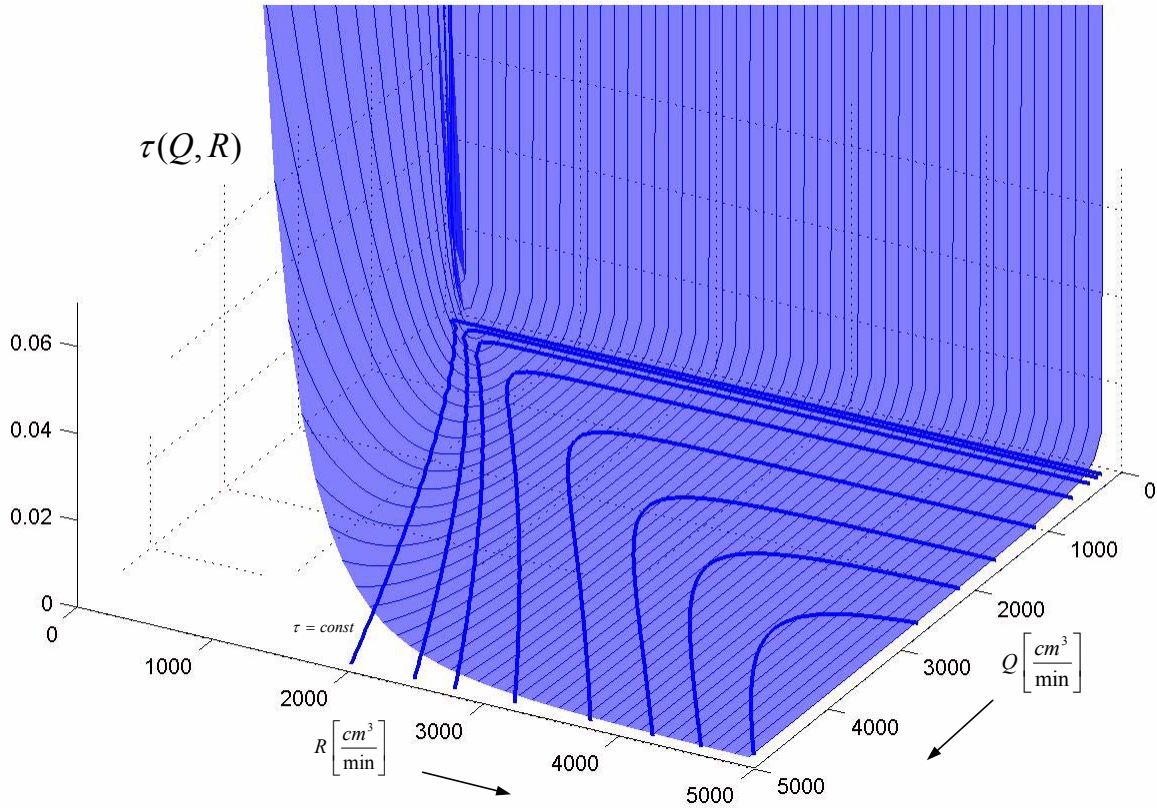
$$Q = \frac{1}{1.358} 270.356 f_z^{0.6} = 199.083 f_z^{0.6}. \quad (57)$$

Intenzitásról forgácsolási paraméterekre áttérve:

$$Q = \frac{d w f}{1000} = \frac{d w f_z z n}{1000} = \frac{d w f_z \cdot z}{1000} \frac{1000 v}{\pi \cdot D_c} = d w f_z z \frac{v}{\pi D_c} = 199.083 f_z^{0.6}, \quad (58)$$

amiből a völgyvonal egyenlete v, f_z síkon:

$$v = \frac{199.083 \pi D_c}{z f_z^{0.4} d w}. \quad (59)$$



25. ábra: A $\tau(R, Q) = \frac{1}{Q} + \frac{Q^4}{R^5} \left[\frac{\text{min}}{\text{cm}^3} \right]$ fajlagos költségkvivalens idő

függvény ábrázolása néhány $\tau = \text{const}$, a Q,R síkra vetített szintvonalával.

Az ábra a MatLab szoftver segítségével készült.

4.6 Az éltartam szórás költségre befolyásolt hatása

4.6.1 A relatív költségeltérés levezetése

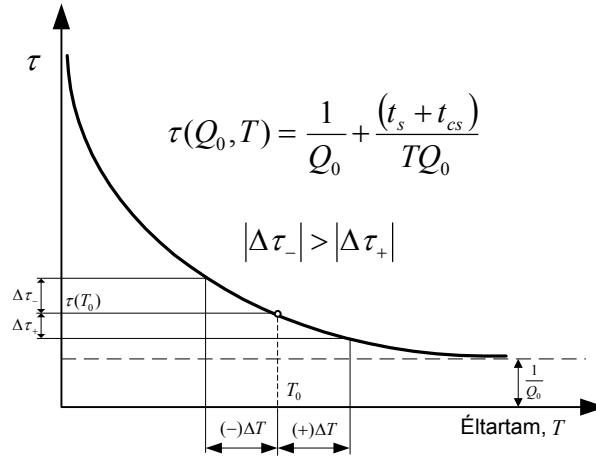
A gyakorlati alkalmazhatóság szempontjából fontosnak tarom, hogy rendelkezésre álljon egy könnyen használható módszer, amely lehetővé teszi az alkalmazhatósági határok egyszerű kezelését. Az itt bemutatott érzékenységi analízis, vagy hibavizsgálat ismert matematikai módszeren alapszik. Hasonló hibavizsgálat található [4]-ben azzal a lényeges különbséggel, hogy az ott bemutatott hibaanalízis csak hagyományos optimalás esetén érvényes. Majd látni fogjuk, hogy az intenzitás alapú optimalás esetében meglepően jó (kompakt) megoldáshoz jutunk.

A forgácsolási folyamat sztochasztikus voltának lényeges jellemzője az éltartam-összefüggés (22) állandóinak $(x_v, y_v, z_v, w_v, p_v)$ a technológiai gyakorlatban adódó szórása. Vizsgáljuk meg, hogy az éltartam szórása milyen hatással van a (32) célfüggvényre. Helyettesítsük be (38)-t a (32) célfüggvénybe ezzel előállítva $\tau(Q, T)$ függvényt:

$$\begin{aligned} \frac{1}{Q} + \frac{Q^{q-1}}{R^q} &= \frac{1}{Q} + \frac{Q^{q-1}}{\left(\left(\frac{T}{(t_s + t_{cs})} \right)^m Q \right)^q} = \frac{1}{Q} + \frac{Q^{q-1}}{\left(\frac{T}{(t_s + t_{cs})} \right)^{mq} Q^q} = \frac{1}{Q} + \frac{(t_s + t_{cs})}{TQ}, \\ \tau(Q, T) &= \frac{1}{Q} + \frac{(t_s + t_{cs})}{TQ}. \end{aligned} \quad (60)$$

Azt fogjuk vizsgálni, hogyan változik a költség, ha a valódi éltartam a sztochasztikus jellege miatt nem az előzetesen előírt T_0 , hanem ettől eltér. Emellett Q -t konstansnak tekintjük ($Q = Q_0$), tehát feltételezzük, hogy már előzetesen meghatároztuk az T_0 -hoz tartozó intenzitást. A 26. ábrán látható a költségekvivalens időfüggvény, mint az éltartamtól függő egyváltozós függvény. Az ábráról leolvasható, hogy az optimális éltartamtól való negatív eltérés $(-)\Delta T$, abszolút értékben nagyobb költségekvivalens idő eltérést okoz, mint a pozitív irányú eltérés $(+)\Delta T$, tehát $(|\Delta \tau_-| > |\Delta \tau_+|)$. A $\tau(Q_0, T)$ T szerinti első deriváltja (61) alapján látható, hogy növekvő éltartam (T), csökkenő meredekséget, kisebb abszolút eltérést okoz:

$$\frac{d\tau(Q_0, T)}{dT} = -\frac{(t_s + t_{cs})}{Q_0 T^2}. \quad (61)$$



26. ábra: a költségekvivalens idő függvény $\tau(Q_0, T)$, mint egyváltozós függvény jelleghelyes ábrázolása. Megállapítható, hogy negatív éltartam eltérés, abszolút értékben nagyobb költségekvivalens idő eltérést okoz.

A mennyiségi vizsgálathoz, képezzük $\tau(Q_0, T)$, mint csak az éltartamtól függő egyváltozós függvény T szerinti Taylor sorát a T_0 helyen:

$$\tau(T) = \tau(T_0) + \frac{\tau'(T_0)}{1!}(T - T_0) + \frac{\tau''(T_0)}{2!}(T - T_0)^2 + \frac{\tau'''(T_0)}{3!}(T - T_0)^3 + \dots \quad (62)$$

Az fajlagos költségekvivalens idő abszolút eltérése:

$$\Delta\tau = \tau(T) - \tau(T_0). \quad (63)$$

Alakítsuk át a (62)-at $\Delta T = T - T_0$ felhasználásával: (ahol ΔT az abszolút éltartam eltérés)

$$\Delta\tau = \frac{\tau'(T_0)}{1!}\Delta T + \frac{\tau''(T_0)}{2!}(\Delta T)^2 + \frac{\tau'''(T_0)}{3!}(\Delta T)^3 + \dots \quad (64)$$

Ha (60) összefüggést behelyettesítjük akkor a következőt kapjuk $\Delta\tau$ -re:

$$\Delta\tau = -\frac{(t_s + t_{cs})}{Q_0 T_0^2} \Delta T + \frac{(t_s + t_{cs})}{Q_0 T_0^3} \Delta T^2 - \frac{(t_s + t_{cs})}{Q_0 T_0^4} \Delta T^3 + \dots + \frac{(t_s + t_{cs})}{Q_0 T_0^{n+1}} \Delta T^n, \quad (65)$$

vezessük be a következő jelölést: $\alpha = \frac{t_s + t_{cs}}{Q_0}$, végezzük el az egyszerűsítő összevonásokat:

$$\Delta\tau = \frac{\alpha}{T_0} \left[-\left(\frac{\Delta T}{T_0}\right) + \left(\frac{\Delta T}{T_0}\right)^2 - \left(\frac{\Delta T}{T_0}\right)^3 + \dots + (-1)^n \left(\frac{\Delta T}{T_0}\right)^n \right],$$

$$\Delta\tau = \frac{\alpha}{T_0} \left[\sum_{i=1}^n (-1)^i \left(\frac{\Delta T}{T_0}\right)^i \right].$$

A relatív fajlagos költségekvivalens idő eltérés meghatározása:

$$\frac{\Delta\tau}{\tau} = \frac{\frac{\alpha}{T_0}}{\frac{1}{Q_0} + \frac{\alpha}{T_0}} \left[\sum_{i=1}^n (-1)^i \left(\frac{\Delta T}{T_0}\right)^i \right] = \frac{(t_s + t_{cs})}{T_0 + (t_s + t_{cs})} \left[\sum_{i=1}^n (-1)^i \left(\frac{\Delta T}{T_0}\right)^i \right], \quad (66)$$

amely tovább egyszerűsítve:

$$\frac{\Delta\tau}{\tau} = \frac{1}{\frac{T_0}{(t_s + t_{cs})} + 1} \left[\sum_{i=1}^n (-1)^i \left(\frac{\Delta T}{T_0}\right)^i \right]. \quad (67)$$

A további vizsgálatokat a $\left| \frac{\Delta T}{T_0} \right| < 1$ esetre korlátozzuk, azaz az éltartamszórásra érvényes a

$-1 < \frac{\Delta T}{T_0} < 1$ feltétel. A (67) szögletes zárójeles kifejezése, ha n tart a végtelenbe, egy konvergens, váltakozó előjelű mértani sor:

$$\sum_{i=1}^n (-1)^i \left(\frac{\Delta T}{T_0}\right)^i = -\frac{\Delta T}{T_0} + \left(\frac{\Delta T}{T_0}\right)^2 - \left(\frac{\Delta T}{T_0}\right)^3 + \left(\frac{\Delta T}{T_0}\right)^4. \quad (68)$$

A mértani sorok összegképlete szerint $a_1 = -\frac{\Delta T}{T_0}$ és $q = -\frac{\Delta T}{T_0}$. Tehát a sor összege:

$$\sum_{i=1}^{n \rightarrow \infty} (-1)^i \left(\frac{\Delta T}{T_0}\right)^i = \frac{a}{1-q} = \frac{-\frac{\Delta T}{T_0}}{1 + \frac{\Delta T}{T_0}}. \quad (69)$$

Behelyettesítve:

$$\frac{\Delta\tau}{\tau} = - \frac{1}{1 + \frac{T_0}{(t_s + t_{cs})}} \frac{\frac{\Delta T}{T_0}}{1 + \frac{\Delta T}{T_0}}. \quad (70)$$

(70) fejezi ki a relatív költségekvivalens idő eltérést általános esetben. Ez az összefüggés érvényes esztergálásra és marásra egyaránt, mivel a kiinduló egyenlet mindkét esetben formailag azonosak, és a végképlet csak megmunkálási módtól függő konstansokat tartalmaz. (70)-ből a következő általános megállapítások következnek:

A relatív költségekvivalens idő eltérés:

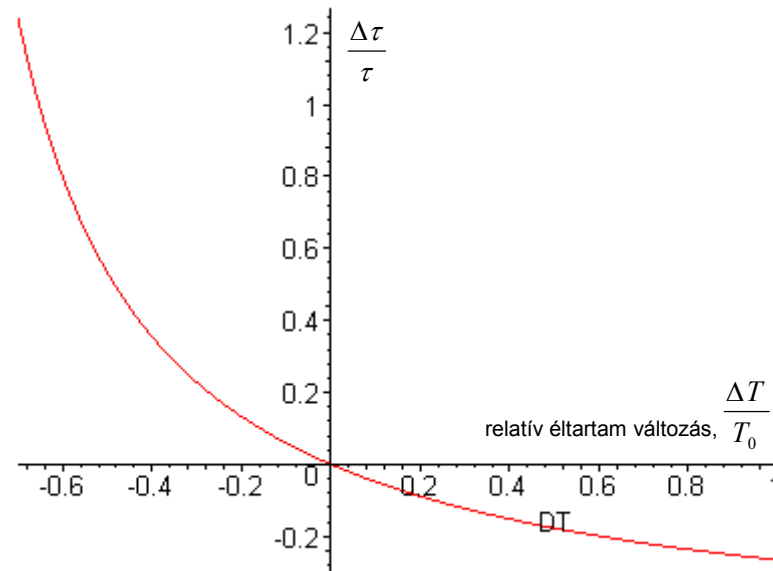
- 1) nem függ az intenzitástól (Q)
- 2) függ az egy él cseréjéhez szükséges átlagos idő (t_{cs}) és a szerszámköltség-ekvivalens időtől ($t_s = \frac{K_{sb}}{k}$) mégpedig úgy, hogy kisebb ($t_s + t_{cs}$) összeg kisebb költség eltérést okoz. A t_s csökken, ha olcsóbb szerszámot használunk, vagy ha azonos szerszámár mellett a környezeti percköltség (k) nő.
- 3) nagyobb T_0 éltartam kisebb relatív költség eltérést okoz, de a T_0 növelése termelékenységi okokból nem célszerű. Minden optimális Q_{opt} optimális intenzitáshoz tartozik egy optimális T_0 éltartam. Kérdésként merülhet fel az, hogy mekkora lesz a relatív eltérés, ha éppen a műveletelemhez tartozó optimális éltartam értékével történik a megmunkálás, azaz a keresési tartomány nem degenerálódik? Erre a kérdésre a következő alfejezet fog választ adni.

Általánosságban elmondható, hogy az elmúlt fél évszázad folyamán az optimálisnak elfogadott éltartam jelentősen csökkent, ami egyértelműen az ipari fejlettséggel magyarázható és ez a tendencia, a műszaki fejlődés pozitív következménye [1,4,16,17]. A fejlődés központi eleme természetesen a szerszámanyag, de jelentős befolyása van a szerszámkonstrukció, a gépkonstrukció és teljesítmény fejlődésének is. Természetesen minden szerszámanyagnál elérhető magas éltartamérték is, az intenzitás csökkentésével, alacsony termelékenység mellett. Téves volna azt hinni, hogy az éltartam optimalitásának kritériuma, annak nagysága, tehát magas éltartam egyben optimális is. Ugyanakkor, ha az éltartam alacsony, az még nem biztos, hogy egyben optimális, mivel az alacsony éltartam lehet a szakszerűtlen üzemeltetés következménye, ami egyben gazdaságtalan szerszámfogyást eredményez.

$\frac{\Delta\tau}{\tau}(t_s, t_{cs}, T_0, \frac{\Delta T}{T_0})$ függvény konstansai, a jelenlegi technikai körülmények mellett a következő számértékeket vehetik fel:

$$t_s = \frac{K_{sb}}{k} = \frac{2500}{166} \approx 15 \text{ min}, t_{cs} = 2 \text{ min}, T_0 = 60 \text{ min}. \quad (71)$$

A függvény grafikonja az 27. ábrán látható $\frac{\Delta\tau}{\tau}$ és $\frac{\Delta T}{T_0}$ koordináta-rendszerben a fenti konstansok figyelembevételével.



27. ábra: (70) ábrázolása egy konkrét példa esetén, ahol $t_{cs} = 2 \text{ min}$, $T_0 = 60 \text{ min}$, $t_s = 15 \text{ min}$. Látható, hogy negatív éltartameltérés nagyobb relatív hibát okoz, mint a pozitív eltérés. Az ábráról leolvasható, hogy 40% negatív éltartameltérés $\approx 36\%$ költséghibát okoz, ugyanakkor 80% pozitív eltérés körülbelül $\approx 22\%$ költséghibával jár.

4.6.2 Egyszerűsítések szabályos keresési tartomány esetén

Már az előzőekben utaltam rá, hogy minden, a modell alapján megállapított Q intenzitáshoz tartozik egy éltartam érték is egyben. Azt fogjuk vizsgálni, milyen speciális helyzetet eredményez az a tény a hibavizsgálat szempontjából, ha a keresési tartomány szabályos, tehát az optimális Q intenzitás meghatároz egy hozzá tartozó optimális éltartamot.

Ha az optimalizációs tartományt szabályosnak tekintjük, tehát a völgyvonal és az optimumvonal egyenesei a tartományon belül maradnak, akkor egy adott műveletelemre az optimumvonal $R = \text{MIN}(\partial\tau/\partial Q)$ egyenesének meredeksége meghatározza az optimális éltartamot:

$$R = \text{MIN}(\partial\tau/\partial Q)$$

$$\frac{\partial\tau}{\partial Q} = \frac{1}{Q^2} + \frac{(q-1)Q^{q-2}}{R^q} = 0; \quad (72)$$

amiből:

$$R = (q-1)^{\frac{1}{q}} Q. \quad (73)$$

Az optimumvonal meredeksége $(q-1)^{\frac{1}{q}}$. Az éltartamkorlát meredeksége (38) alapján

$C_T = \left(\frac{T_0}{t_s + t_{cs}} \right)^{\frac{1}{q}}$. Tehát a szabályos keresési tartomány esetén ezt a két meredekséget egyenlővé tehetjük.

$$\left(\frac{T_0}{t_s + t_{cs}} \right)^{\frac{1}{q}} = (q-1)^{\frac{1}{q}}, \quad (74)$$

ebből T_0 kifejezhető:

$$T_0 = (t_s + t_{cs}) (q-1) \quad (75)$$

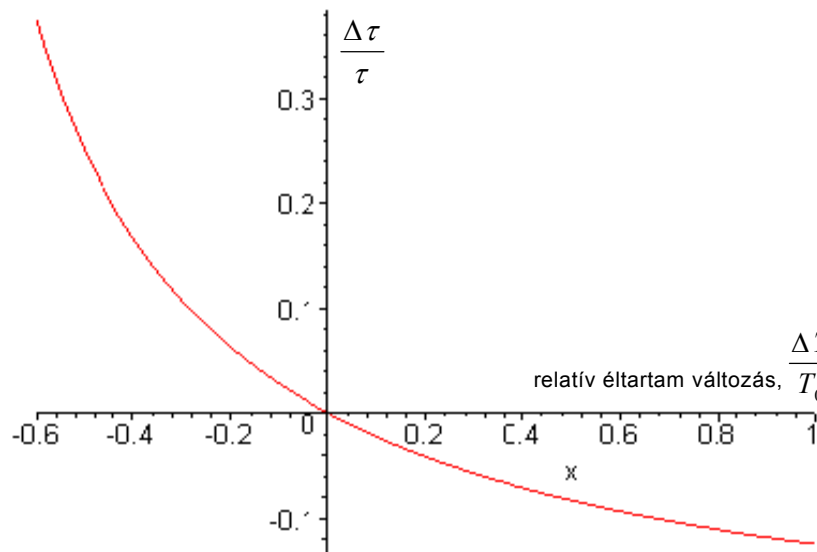
(70) be visszahelyettesítve:

$$\frac{\Delta\tau}{\tau} = - \frac{1}{1 + \frac{(t_s + t_{cs})(q-1)}{(t_s + t_{cs})}} \frac{\frac{\Delta T}{T_0}}{1 + \frac{\Delta T}{T_0}}, \quad (76)$$

$$\frac{\Delta \tau}{\tau} = -\frac{1}{q} \frac{\frac{\Delta T}{T_0}}{1 + \frac{\Delta T}{T_0}}. \quad (77)$$

(77) alapján kijelenthetjük, ha a keresési tartomány szabályos, akkor a relatív költségeltérés nem függ t_s , t_{cs} -től, hanem csak a Taylor éltartamegyenlet m kitevőjének reciprokától ($q = \frac{1}{m}$) és m egyenesen arányos a relatív költségeltéréssel: amennyiben m csökken, a relatív költségeltérés is csökken. Nagyoló esztergálás esetén a (77)-el megegyező összefüggés vezethető le. A $\frac{\Delta \tau}{\tau} \left(\frac{\Delta T}{T_0} \right)$ függvény grafikonja a 28. ábrán látható, $q = 4$ behelyettesítésével.

Az ábra mutatja, hogy ebben az esetben az éltartam sztochasztikus volta kevésbé befolyásolja a műveletelem költségét, mint olyan esetekben, ha nem tudjuk az optimális intenzitást alkalmazni, a keresési tartomány degenerációja miatt.



28. ábra: (77) ábrázolása egy konkrét példa esetén, ahol $q = 4$ (a technológiai gyakorlatban a q 3 és 5 közötti valós szám). Látható, hogy negatív éltartameltérés nagyobb relatív hibát okoz, mint a pozitív eltérés. Az ábráról leolvasható, hogy 40% negatív éltartameltérés $\approx 18\%$ költséghibát okoz, ugyanakkor 100% pozitív eltérés (!), körülbelül $\approx 12\%$ költséghibával jár.

4.6.3 A hibavizsgálat összefoglalása

Ha a műveletelemhez tartozó *optimális intenzitás* Q_{opt} szabályos keresési tartományban adódott, akkor a relatív éltartam változás, sokkal kedvezőbb befolyással van a relatív költség ekvivalens idő változásra, mintha Q_{opt} a degenerált keresési tartományban adódott volna.

A hibavizsgálat fő célja az, hogy segítségével megadhatjuk a megengedett relatív költség ekvivalens idő eltérés előírásakor a megengedett éltartamszórás tartományát is. Amennyiben ez a szórási tartomány *nagyobb*, mint amely a vizsgált gyártási feladatra a technológiai gyakorlatban valójában előfordul, akkor a *statikus optimalizálás* helyes eredményt ad. Ellenkező esetben viszont, *dinamikus* optimalizáló módszereket kell alkalmazni.

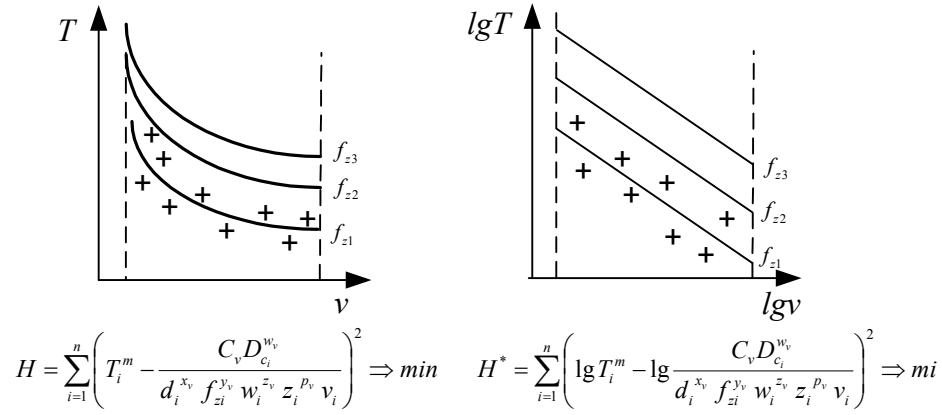
4.7 Minta – adatbázis létrehozása

A fejezetben bemutatott optimalizációs eljárás akkor alkalmazható a gyakorlatban, ha rendelkezésre áll egy adatbázis, amely az éltartamegyenlet konstansait tartalmazza. Esztergálás esetére a [14] mellékletében találhatunk egy adatbázist, táblázatos formában. Ehhez hasonlóan érdemes kialakítani a megfelelő marásra is.

A minta-adatbázis létrehozása a TITEX katalógusának adatai alapján történt. A (22) éltartamegyenlet $C_v, x_v, y_v, z_v, w_v, p_v, m$ állandóit a rendelkezésre álló $T_i, D_{ci}, d_i, f_{zi}, w_i, z_i, v_i$ értékekhez több-változós nemlineáris regresszió segítségével úgy kell meghatározni, hogy a hibanégyzeteknek minimuma legyen:

$$H = \sum_{i=1}^n \left(T_i^m - \frac{C_v D_{ci}^{w_v}}{d_i^{x_v} f_{zi}^{y_v} w_i^{z_v} z_i^{p_v} v_i} \right)^2 \Rightarrow \min. \quad (78)$$

A H hibafüggvény szélsőértékének elvi meghatározása a következő: első lépésben előállítjuk a hibafüggvény keresett változói szerinti parciális deriváltjait, majd nullával egyenlővé téve a kapott egyenleteket, 7 ismeretlenes nemlineáris egyenletrendszer kell megoldani. A kapott egyenletekből a keresett állandók kifejezése explicit formában nem lehetséges. Ezért az ilyen feladatok megoldásához közelítő módszereket alkalmaznak, például lineáris regressziót, amely kihasználja a keresett hatványfüggvény azon előnyös tulajdonságát, hogy logaritmikus koordináta-rendszerben egyenes.



29. ábra: az éltartamfüggvény meghatározása többváltozós lineáris regresszióval.

Alkalmazzuk H közelítésére a következő összefüggést:

$$H^* = \sum_{i=1}^n \left(\lg T_i^m - \lg \frac{C_v D_{c_i}^{w_v}}{d_i^{x_v} f_{z_i}^{y_v} w_i^{z_v} z_i^{p_v} v_i} \right)^2 \Rightarrow \min. \quad (79)$$

A H^* minimumának meghatározását a következőképpen végezhetjük: Először végezzük el az összes keresett változók szerinti parciális deriválásokat. Hét darab egyenletet kapunk hét ismeretlennel. H^* minimuma ott lehet, ahol a parciális deriváltak zérussal egyenlők. A levezetéseket a 1. melléklet tartalmazza. A levezetés eredménye egy hét ismeretlenes lineáris egyenletrendszer, amelynek együttható mátrixa csak konstans elemeket tartalmaz. Az egyenletrendszer legszemléletesebb megadása a következő mátrix-vektor szorzat:

$$\begin{bmatrix} \sum_{i=1}^n \lg T_i & -\sum_{i=1}^n \lg D_{c_i} & \sum_{i=1}^n \lg d_i & \sum_{i=1}^n \lg f_{z_i} & \sum_{i=1}^n \lg w_i & \sum_{i=1}^n \lg z_i & -n \\ \sum_{i=1}^n \lg T_i \lg d_i & -\sum_{i=1}^n \lg D_{c_i} \lg d_i & \sum_{i=1}^n (\lg d_i)^2 & \sum_{i=1}^n \lg f_{z_i} \lg d_i & \sum_{i=1}^n \lg w_i \lg d_i & \sum_{i=1}^n \lg z_i \lg d_i & -\sum_{i=1}^n \lg d_i \\ \sum_{i=1}^n \lg T_i \lg f_{z_i} & -\sum_{i=1}^n \lg D_{c_i} \lg f_{z_i} & \sum_{i=1}^n \lg d_i \lg f_{z_i} & \sum_{i=1}^n (\lg f_{z_i})^2 & \sum_{i=1}^n \lg w_i \lg f_{z_i} & \sum_{i=1}^n \lg z_i \lg f_{z_i} & -\sum_{i=1}^n \lg f_{z_i} \\ \sum_{i=1}^n \lg T_i \lg w_i & -\sum_{i=1}^n \lg D_{c_i} \lg w_i & \sum_{i=1}^n \lg d_i \lg w_i & \sum_{i=1}^n \lg f_{z_i} \lg w_i & \sum_{i=1}^n (\lg w_i)^2 & \sum_{i=1}^n \lg z_i \lg w_i & -\sum_{i=1}^n \lg w_i \\ \sum_{i=1}^n \lg T_i \lg z_i & -\sum_{i=1}^n \lg D_{c_i} \lg z_i & \sum_{i=1}^n \lg d_i \lg z_i & \sum_{i=1}^n \lg f_{z_i} \lg z_i & \sum_{i=1}^n \lg w_i \lg z_i & \sum_{i=1}^n (\lg z_i)^2 & -\sum_{i=1}^n \lg z_i \\ \sum_{i=1}^n \lg T_i \lg D_{c_i} & -\sum_{i=1}^n (\lg D_{c_i})^2 & \sum_{i=1}^n \lg d_i \lg D_{c_i} & \sum_{i=1}^n \lg f_{z_i} \lg D_{c_i} & \sum_{i=1}^n \lg w_i \lg D_{c_i} & \sum_{i=1}^n \lg z_i \lg D_{c_i} & -\sum_{i=1}^n \lg D_{c_i} \\ \sum_{i=1}^n (\lg T_i)^2 & -\sum_{i=1}^n \lg D_{c_i} \lg T_i & \sum_{i=1}^n \lg d_i \lg T_i & \sum_{i=1}^n \lg f_{z_i} \lg T_i & \sum_{i=1}^n \lg w_i \lg T_i & \sum_{i=1}^n \lg z_i \lg T_i & -\sum_{i=1}^n \lg T_i \end{bmatrix} \begin{bmatrix} m_v \\ w_v \\ x_v \\ y_v \\ z_v \\ p_v \\ \lg C_v \end{bmatrix} = \begin{bmatrix} -\sum_{i=1}^n \lg v_i \\ -\sum_{i=1}^n \lg v_i \lg d_i \\ -\sum_{i=1}^n \lg v_i \lg f_{z_i} \\ -\sum_{i=1}^n \lg v_i \lg w_i \\ -\sum_{i=1}^n \lg v_i \lg z_i \\ -\sum_{i=1}^n \lg v_i \lg D_{c_i} \\ -\sum_{i=1}^n \lg v_i \lg T_i \end{bmatrix} \quad (80)$$

A (80)-ben szereplő ($\mathbf{Ax} = \mathbf{b}$) alakú egyenletrendszer ($\mathbf{x}$ jelenti az ismeretlen vektort) megoldható valamilyen numerikus módszer segítségével. A legismertebb *Cramer* szabály alkalmazása [34], nem vezet pontos eredményre mert az együttható mátrix kondíciószáma nagy. A MatLab szoftvercsomag segítségével több ismert eljárást teszteltem, és a MatLab beépített \ operátorával előállított eredmény adta a legkisebb reziduális hibát. A mérési eredmények

bevitelére a Microsoft Excel szoftvert használtam fel, ami kétség kívül a legkényelmesebb adatkezelést biztosítja, de a számítást az Excel Link ToolBox segítségével a MatLab végezte. Ez a Matlab Toolbox egy úgynevezett 'Add-in', amely egyszerű adatkapcsolatot képes létesíteni a két szoftver között. A minta-adatbázis több száz mérés eredményét tartalmazza táblázatos formában (lásd 2. számú függelék). A táblázat első két oszlopa a szabványos munkadarab anyagcsoportokat és ezen belül a lehetséges alcsoportokat tartalmazza. Az egyes szerszámazonosítók meghatározzák a szerszám átmérőjét, forgácsolási sebesség, előtolás, fogásmélység és szélesség határait. A táblázat utolsó hét oszlopa a Taylor éltartam egyenlet 6 kitevőjét és egy konstans szorzóját tartalmazza.

5 Forgácsolási erő modellezése megmunkálás-szimulátorokban

5.1 Bevezetés

A megmunkálás gazdaságossága, minősége és a forgácsolási paraméterek közvetlen kapcsolatban vannak. Olyan feltételeket keresünk, amelyek mellett, biztonságos forgácsolási viszonyok és minimális megmunkálási idő mellett megfelelő felületminőség kritériuma is teljesülhet. A biztonságos forgácsolási viszonyok és a felületi minőség egyaránt a forgácsoláskor fellépő erőktől függnék. Ezért érdemes szimulátorokat fejleszteni és alkalmazni, amelyek lehetővé teszik a megmunkáláskor fellépő erők meghatározását is. A forgácsoláskor fellépő erők ellenőrzésével biztosítható hogy nem lépjük túl a megengedett főorsó teljesítményt, a forgácsoló él nem fog eltörni, és a szerszámszárra ható hajlító erők nem fognak szerszámtörést okozni és végül de nem utolsó sorban, a szerszám elhajlása nem fog egy előre megadott mértéket túllépni.

A szoborszerű felületek megmunkálásának nehézségét az adja, hogy a forgácsolási viszonyok folyamatosan változnak a különböző műveleteken és műveletelemeken belül, ami elsősorban a munkadarabok változatos geometriájával magyarázható. A 2 és fél tengelyes megmunkálások esetén, pl. zsebmaráskor még elérhető, hogy a fogásmélység és a fogásszélesség egy műveletelemen belül állandó, de valódi 3 tengelyes fél-simító és simító műveletek esetén a forgácsolási paraméterek folyamatosan változnak. Ennek oka az, hogy a nagyolás az esetek nagy részében teraszos jellegű ráhagyási felületet alakít ki és ennek lesimításakor a fogásmélység és ezzel együtt a forgácsolási erő, a ráhagyási alakzat (felület) függvénye. A ráhagyási felület modellezése nem egyszerű feladat, a marás-szimulációval foglalkozó rész ezt részletesen tárgyalta. Ha sorozatgyártásról van szó, akkor a szerszámpálya optimalizálható a tapasztalat alapján is, de a szoborszerű felületek általában egyedi alkatrészekként kezelendők, ezért az ilyen tapasztalat ezeknél nem ad kielégítő eredményt.

Mint már említettem, a pontos forgácsolási erő meghatározás nem csak gazdaságossági hatással bír, hanem emellett a komoly befolyása van a kész alkatrész minőségére is. A forgácsoláskor fellépő túl nagy forgácsolási erők a szerszámot elhajlítják, és nagyobb alak vagy helyzeteltérés keletkezhet, mint amit a tűrésmező megenged. A forgácsolási erő nagysága természetesen a szerszámkopással is arányos, a tapasztalat szerint a változó erőviszonyok sokszor előre becsülhetetlen szerszám éltartamot eredményeznek. A továbbiakban kizárólag a marásra alkalmazható modellekre fogunk koncentrálni részletesen kitérve a térfogat és mechanikai modellekre.

5.2 Marás

A disszertáció elején bemutatott 19. ábrán a homlokmarás geometriai viszonyai láthatóak. Az előtolás f irányába mozgó szerszám $V_m = d \cdot w \cdot l_w$ térfogatú anyagot távolít el a munkadarabból. Mivel a forgácsoló él nem egyenes pálya, hanem hurkolt ciklois mentén mozog, ezért a forgácsvastagság $h(\varphi)$ szerint folyamatosan változik, ahol φ az él pillanatnyi szöghelyzetét jelöli. A pillanatnyi forgácsvastagság függ az él szöghelyzetétől és az előtolás nagyságától egyaránt. Az előtolás pedig felírható a fogankénti előtolás, a fogszám és a fordulatszám segítségével (lásd: alapképletek). A forgácsolási erő függ a radiális és axiális fogásmélységtől.

A forgácsolási erőt befolyásoló független változókat a következő táblázat foglalja össze, a munkadarab, szerszám és szerszámgép szerinti csoportosításban:

Munkadarab	Szerszám	Szerszámgép
<i>anyaga</i> - szakitó szilárdság, keménység, alakíthatóság	<i>anyaga</i> - keménység, szívósság, hőellenállás, bevonat típusa <i>forgácsolás geometria</i> - fogásmélység, fogásszélesség <i>forgácsolási feltételek</i> - főorsó fordulatszám, fogankénti előtolás, hűtő-kenő folyadék típusa	<i>gépjellemzők</i> - merevség, szerszám excentricitás a befogásnál, teljesítmény-, fordulatszámkorlátok

1. táblázat: A forgácsolási erőt befolyásoló független változók munkadarab, szerszám és gép szerinti csoportosításban. Ezek a paraméterek konstansként szerepelnek a legtöbb modellben.

Fontos megjegyezni, hogy a felsorolt változók kis megváltozása is jelentősen befolyásolhatja a forgácsolási erőt és e mellett a szerszámgép egyes jellemzőit is sokszor nagyon nehéz előre számszerűleg becsülni. (A jelen dolgozatban bemutatott modellek ezeket a jellemzőket elhanyagolják.) Már sokszor említést tettem függő változókról, amelyeket a táblázatban felsorolt paraméterek befolyásolnak, ezért most összefoglalásként soroljuk fel a négy legfontosabbat ezek közül:

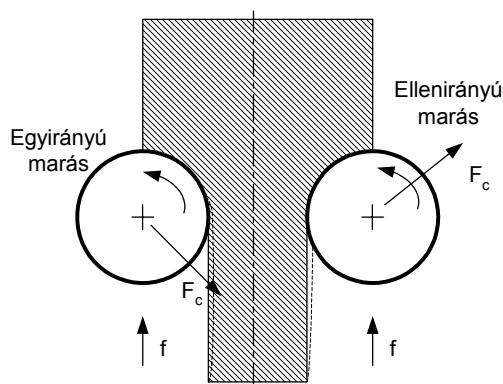
- gazdaságosság - megmunkálási idő, szerszámkopás
- minőség - mérettűrés, felületminőség
- forgácsolási erő – az egy fogra ható erő, és erővektor abszolút értéke és iránya

- szerszám elhajlása - iránya és abszolút értéke.

A megmunkálás-szimulátorok fontos feladata e négy paraméter meghatározása, az adott független változók pillanatnyi értéke alapján. Ha valamilyen nem megengedett eltérést mutat ez a vizsgálat, akkor még a megmunkálás megkezdése előtt informálni lehet a felhasználót a felmerülő problémákról.

A forgácsolási paraméterek hagyományos megválasztása technológusi kézikönyveken alapul [57,56] az ezekben fellelhető nagy mennyiségű táblázatokból kiolvasható az adott szerszám-munkadarab párhoz egy ajánlott forgácsolási sebesség. Ez a megközelítés meglehetősen jól működik, ha a forgácsolási paraméterek a munkadarab geometriája miatt nem változnak meg jelentősen. Általánosságban elmondható, hogy első közelítésben érdemes ezekkel a kiinduló adatokkal dolgozni, és a későbbiekben a változó feltételekhez igazítani őket.

A különböző megmunkálási fázisok eltérő követelményekkel rendelkeznek. Simításkor a pontosság kerül előtérbe, a szerszámmra ható erő nem haladhat meg egy bizonyos előre megadott korlátértéket, mivel az elhajlás, rossz felületminőséget, a tűrésmezőből kilépést okozhat. Nagyoláskor a forgácsoló erő abszolút értékének kisebbnek kell lennie a fogtörést okozó erőnél. A szerszámkopás és a rezgések becslése pedig minden fázisban nagy jelentőséggel bír.



30. ábra: ellenirányú és egyirányú maráskor keletkező F_c erő iránya és nagysága megegyezik, de ellenkező irányú elhajlást okoz a munkadarabhoz képest. Ez az oka, hogy a gyakorlatban túlnyomórészt az ellenirányú marást alkalmazzák.

A 30. ábra illusztrálja szerszámelhajlást a két legegyszerűbb marási változat esetén [54] a szerszámelhajlás ellenőrzésének fontosságát. Az úgynevezett zig-zag szerszámpálya alkalmazása esetén a szerszám felváltva dolgozik egy- és ellenirányban. Nagyobb fogásmélység vagy fogásszélesség növeli az F_c erőt, ami nagyobb elhajlást okoz. Mivel az elhajlásvektor iránya "zig" és "zag" menetben eltérő, ezért ez megduplázza az eltérés

tartományát. Ez magyarázza azt az általános gyakorlatot, hogy simításkor mindig egy-utas ("one-way") szerszámpályát alkalmaznak főként ellenirányú változatban [54].

Vizsgáljuk meg tehát azt a kérdést, hogyan lehet ezt az elhajlást szabályozni?

Ehhez a következő lépések szükségesek:

2. Alkalmazni kell egy olyan geometriai modellt, melynek segítségével a forgácsolási folyamat alatt tetszőleges időpillanatban meg tudjuk határozni az úgynevezett érintkezési tartományt "contact area" a szerszám felületén, vagy egy előre megadott síkra vetítve.
3. Meg kell határozni az erővektor irányát és nagyságát az érintkezési tartomány vizsgálatával.
4. Modellezni kell a szerszámelhajlást a kiszámított erő alapján.
5. Meg kell határozni az elhajlás mértékét. Ha az elhajlás nagyobb, mint az adott tűrésmező által megengedett, akkor az előtolást addig kell csökkenteni, amíg ez nem haladja meg a megengedett értéket.

Az utolsó lépésben alternatívaként fel lehet ajánlani más bejárési stratégiákat is, ha az előtolás csökkentése technológiai okokból nem lehetséges. Ilyenkor a szimulátorok jelezhetik az alternatív szerszámpálya szükségességét. Természetesen ez a rugalmas alternatíva csak akkor alkalmazható gyorsan és hatékonyan, ha az ellenőrző szimulátor egy CAM rendszerbe integrált modulként implementáltak. Mint már említettem, ez az integrációs törekvés figyelhető meg a VeriCUT megmunkálás-szimulátor és a Pro/Engineer 2001 CAD/CAM szoftvercsomagok esetén. [32,107]

5.2.1 Térfogat modellek

A forgácsoló erő meghatározásával folyamattervezési, felület minőségi és a szerszámtörés meghatározása szempontjából számos irodalmi hivatkozást találunk. [63,58] Ezek a tanulmányok sok esetben az élhorony geometriájával és az e mentén fellépő erő számításával, mérésével is foglalkoznak. De most először vizsgáljuk meg a legegyszerűbb lehetőséget, a térfogatmodellt, ami a korábbi fejezetekben már tárgyalt anyagleválasztási intenzitás fogalmára alapul, a következő alapegyenlet szerint:

$$P = k Q, \quad [\text{kW}] \quad (81)$$

ahol a k konstans a szerszámkatalógusokból meghatározható minden szerszám és munkadarabanyag párra. A teljesítményből, a tangenciális és a radiális erőkomponensek is számíthatóak:

$$F_t = \frac{kQ}{v}, F_r = k_r F_t, \quad (82)$$

ahol k_r ugyancsak anyag, szerszám és a megmunkálás jellegétől függő konstans. A térfogat modell ez által implementálható, egyetlen nehézséget az intenzitás meghatározása jelenthet. A szimulációnál jól használható *Z-map* modell alkalmas az intenzitás meghatározására, mivel a következő képlet szerint a *Z* irányú tűk hosszváltozása megadja egy szerszámelmozdulás következtében eltávolított anyagtérfogatot:

$$V_m = T_{u,v \rightarrow x,y} \sum_i (z_b - z_a)_i. \quad [\text{cm}^3] \quad (83)$$

A képletben szereplő z_b és z_a jelenti az szerszámelmozdulás előtti és utáni tűmagasságokat, amiket összegezni kell minden tűre. Mivel a *Z-map* egy skálázott u,v koordináta rendszerben dolgozik, ezért a végeredményt be kell szorozni a $u,v \rightarrow x,y$ transzformáció konstansával.

Az intenzitás kiszámításához természetesen el kell osztani a szerszámelmozdulás idejével:

$$Q = \frac{V_m}{t_m} = \frac{V_m}{L_m/f}. \quad [\text{cm}^3/\text{min}] \quad (84)$$

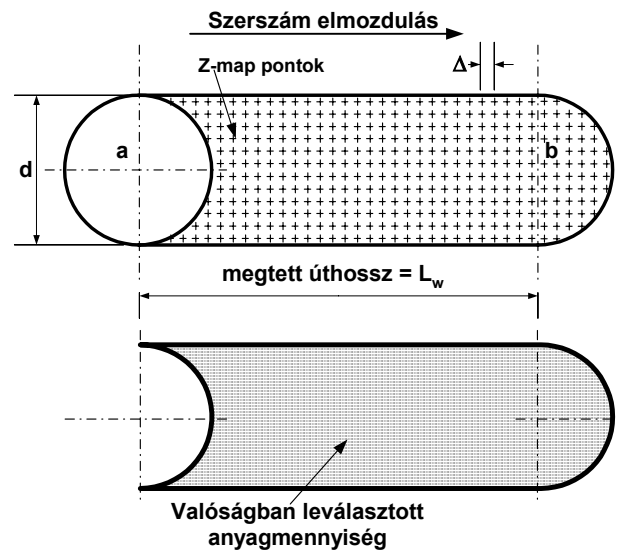
A maximális hibát a következő képlettel lehet számolni hagyományos esetben (31. ábra):

$$\text{hiba} = \frac{\text{nem érzékelt terület}}{\text{valóságban leválasztott terület}} = \frac{(d\pi + L_w)\Delta}{L_w 2d}. \quad (85)$$

Ha a *Z-map* az u,v diszkrét terület felett van akkor a maximális hiba:

$$\text{hiba} = \frac{(UVtoX UVtoY)L_w}{L_w 2d} = \frac{UVtoX UVtoY}{2d}. \quad (86)$$

Az *UVtoX* és *UVtoY* konstansok a *p-buffer* eljárásnál bemutatott $UV \rightarrow XY$ transzformáció konstansai. A CGTech által megvalósított VeriCut szoftver [107], valószínűleg ezt a módszert alkalmazza, bár mint minden kereskedelmi szoftver ez sem ad közre technikai részleteket a működésről. Nyilvánvaló nagy előnye a módszernek az hogy a szerszám nagy előtolással fog mozogni, ha a levegőben mar. A CGTech leírásában közlik, hogy ezzel az egyszerű megoldással is átlagosan 50%-os megmunkálási időcsökkentést lehet elérni.



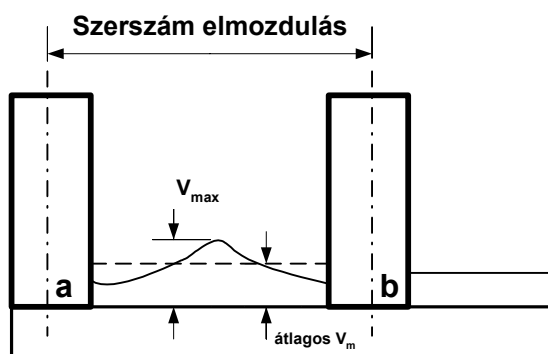
31. ábra: szerszámmozgás A pontból B pontba a z-map felett. Ez a módszer egy közelítő megoldást ad a leválasztott anyag térfogatára, mivel csak a kis keresztekkel jelölt diszkrét pontokban mérjük az anyagleválasztást

A táblázatok később folyamatosan módosíthatóak, finomíthatóak az újabb anyagok és szerszámok adataival. Nagyolásra, félsimításra használható táblázatok nem alkalmazhatóak simításra, ahol a fellépő erőket és elhajlásokat kell kontrolálni. Az alapvető hiányossága a térfogatmodellnek az, hogy a pillanatnyi erő számítására nem alkalmas. Az erő iránya és ezért az elhajlás iránya is ismeretlen, így a felületi minőség ellenőrzése sem lehetséges. (32. ábra) A módszer gyakorlatilag átlagos teljesítményszükségletet számol, amit nagyoláskor természetesen mindig célszerű meghatározni. Ha mégis pillanatnyi teljesítmények szükségesek, akkor a hiányosságokat könnyen ki lehet küszöbölni a sűrűbb mintavételezéssel, azaz 'a' és 'b' között több pontban vizsgálhatjuk a leválasztott anyagmennyiséget, és ha az átlagtól nagy eltérést észlelünk, érdemes a szakaszt felosztani két szakaszra eltérő előtolás alkalmazása mellett.

5.2.2 Mechanikai modellek

A [59] -ban bemutatott mechanikai modell a térfogatmodell fő hiányosságát oldja meg, azaz képes az erő irányának a meghatározására is. Az előző részben részletesen ismertetett térfogat modellt Kline és DeVor [60] terjesztette ki azzal, hogy a szerszámot felosztotta diszkrét, axiális elemekre. Gömbvégű marók esetére Ersoy [61] fejlesztett ki dinamikus módszert, aki gyakorlati megmunkálásokban is ellenőrizte a modell helyességét. A módszer kulcsa az, hogy meg kell határozni, hogy a szerszám mely része vesz részt a pillanatnyi anyagleválasztásban. A mechanikai modell CAPP rendszerben való felhasználásával Spence és Altintas [62] foglalkozott. A módszert ki lehet terjeszteni a szerszámkopás és dinamika

vizsgálataira is. A következőkben bemutatom a módszer lényegét és végül alkalmazási lehetőségeit általában és a saját fejlesztésű szimulátorban.



32. ábra: az általános térfogatmodell hiányossága az, hogy az eredő forgácsolási erő számítása nem pillanatnyi, hanem átlagolt anyagleválasztási intenzitás alapján történik, tehát az erők pillanatnyi maximuma helyett, csak az átlag forgácsolási erő nagyságát lehet meghatározni, irányát nem.

5.2.3 A mechanikai modell megvalósítása

Két fontos jellemzőt kell meghatározni a modell megértéséhez és megvalósításához. Az egyik a *pillanatnyi forgácsvastagság*, amely a szerszám és a munkadarab geometriájától függ, és gyors meghatározása nagymértékben növeli a módszer hatékonyságát. A másik az *érintkezési tartomány*, amely a szerszám felületén rajzolódik ki megmunkálás közben és megadja azt, hogy a forgó szerszám mely pontjai vesznek részt a pillanatnyi forgácsképzésben. Majd látni fogjuk, hogy az érintkezési tartomány egy előre megadott síkra is vetíthető, a hatékonyság növelése érdekében.

5.2.3.1 A pillanatnyi forgácsvastagság

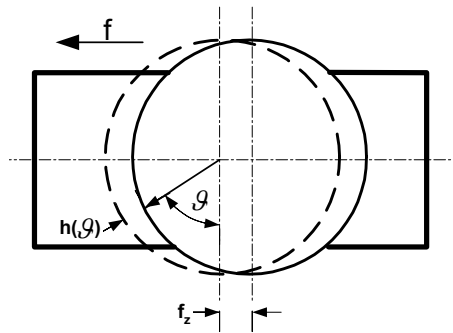
Pillanatnyi forgácsvastagság a ϑ szög függvénye, az előtolás irányába mozgó szerszám esetében: (33. ábra)

$$h(\vartheta) = f_z \sin(\vartheta) = \frac{f \sin(\vartheta)}{z n}, \quad (87)$$

amely egyszerű 2 tengelyes megmunkálás esetén hatékony, de például gömbvégű szerszám esetén, vagy 5 tengelyes megmunkálás esetén a következő képlettel lehet számolni:

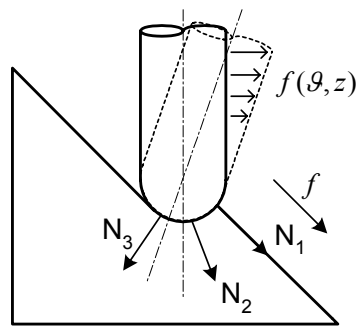
$$h(\vartheta, z) = \frac{f(\vartheta, z) \bullet N(\vartheta, z)}{z \cdot n}. \quad (88)$$

Ilyenkor az előtolás $f(\vartheta, z)$ és a szerszám felületi normális $N(\vartheta, z)$ skaláris szorzata adja a forgácsvastagságot minden pontban a szerszám felületén. 3 tengelyes megmunkálásnál $f(\vartheta, z)$ konstans vektor, de 5 tengelyes esetben változik a szerszám szögelfordulása miatt. (34. ábra)



33. ábra: pillanatnyi forgácsvastagság a ϑ szög függvénye, az előtolás irányába mozgó szerszám esetében.

A (88)-ben szereplő skaláris szorzat lehet negatív is, ami azt jelenti, hogy az adott pont a szerszám hátsó oldalán van, és nem vesz részt a forgácsképzésben.

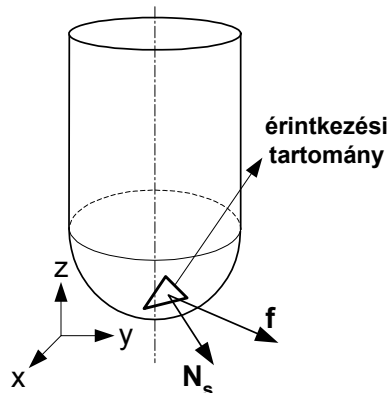


34. ábra: a valóságos forgácsvastagság függvénye a felületi normális és az előtolás vektoroknak.

5.2.3.2 Az érintkezési tartomány

Érintkezési tartománynak (*contact area*) nevezzük azon pontok halmazát a szerszám felületén, amelyek pillanatnyilag részt vesznek a forgácsképzésben. (Ezeket a pontokat az x-y síkra levetítve egy körlapon szemléletesen lehet ábrázolni) Az OpenGL alapú szimulátor minden szerszámhelyzetben könnyen meghatározza ezeket a pontokat az úgynevezett stencil-

buffer segítségével. Az érintkezési tartomány gyors meghatározása kulcsfontosságú a forgácsoló erő modellezésénél.



35. ábra: érintkezési tartománynak nevezzük a modellben azon pontok halmazát a szerszám származtató felületén, amelyek pillanatnyilag részt vesznek a forgácsképzésben. (ezeket a pontokat az X-Y síkra vetítve egy körlapon szemléletesen lehet ábrázolni)

5.2.4 A forgácsolóélre ható erő meghatározása

A hagyományos gömbvégű szármaró modellje a 36. ábrán látható, ahol ϑ jelenti a szerszám elfordulási szögét. A forgácsolás közben az élre vagy lapkára ható erő egyenesen arányos a *forgács keresztmetszettel*. Az erő érintőleges és sugárirányú komponensei (Tlustý, 1975):

$$\begin{aligned}\Delta F_t &= K_t h(\varphi, z) \\ \Delta F_r &= K_r \Delta F_t.\end{aligned}\tag{89}$$

Az előző képletben szereplő K_t és K_r függ a szerszám és munkadarab anyagától, a hőmérsékletétől, a szerszám geometriájától. A szerszám élén fellépő erő függ, a szerszám elfordulás szögétől és a csavarvonal menetemelkedési szögétől. Kifejezetten gömbvégű marószerszám esetére számos kísérlet található a szakirodalomban: [63,64,65,66,67,68,69]. Ebben a speciális esetben (89) a következő formában írható fel:

$$\begin{aligned}dF_T &= K_T(\varphi) h(\vartheta, \varphi)^{m_T} dz \\ dF_R &= K_R(\varphi) h(\vartheta, \varphi)^{m_R} dz,\end{aligned}\tag{90}$$

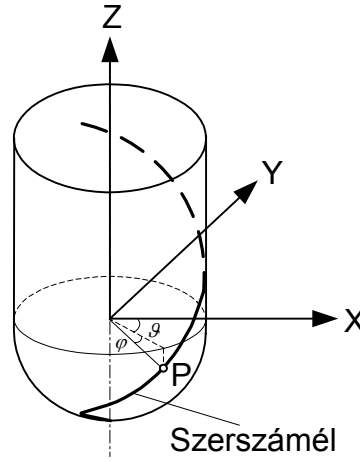
ahol $0 < m_R < 1$ és $0 < m_T < 1$ anyagfüggő állandók, K_T és K_R viszont függvénye a φ szögnek.

Gyakorlati mérésekkel igazolható, hogy $K_R(\varphi)$ maximuma $\varphi = \frac{\pi}{4}$ szögnél van és

$K_T(\varphi)$ monoton csökkenő függvénye φ -nek [64]. Az utóbbi tanulmány szerint, ahhoz hogy megfelelő pontossággal meg tudjuk határozni ezeket a fizikai tényezőket, a továbbiakban a következő közelítő képleteket célszerű alkalmazni:

$$\begin{aligned} K_T(\varphi) &= a_0 + a_1\varphi + a_2\varphi^2 + a_3\varphi^3 \\ K_R(\varphi) &= b_0 + b_1\varphi + b_2\varphi^2 + b_3\varphi^3 \end{aligned} \quad (91)$$

A képletben szereplő 8 darab konstans próbamarások sorozatával minden anyag-szerszám kombinációra megadható. A jelen dolgozat keretében gyakorlati méréseket nem végeztem.



36. ábra: a gömbvégű marószerszám modellje. A szerszámél egy tetszőleges P pontja kifejezhető φ, θ szögekkel.

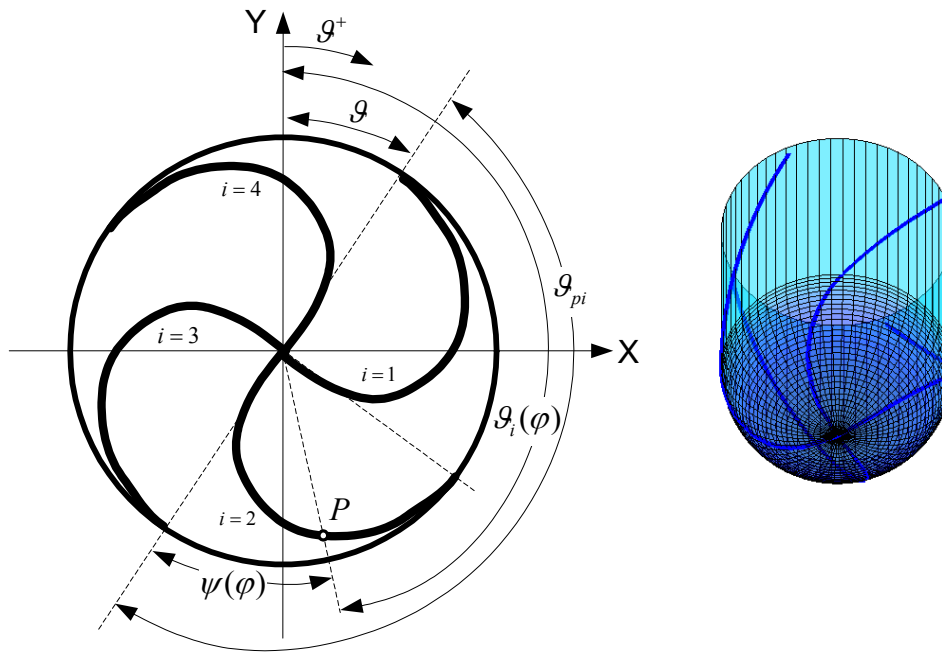
A 37. ábrán látható a modellben használt szögek geometriai rendszere, amelyben feltüntetett szögek a következők: θ a szerszám szögelfordulása, φ az élelem helyzeti szöge, θ_{pi} az i . szerszámélhez tartozó csavarvonal kezdőszöge, $\psi(\varphi)$ a P pont fázisszöge. Ezek alapján az elemi éldarab $\theta_i(\varphi)$ elfordulási szögét 37. ábra alapján a következő képlettel számíthatjuk [68]:

$$\theta_i(\varphi) = \theta + \theta_{pi} - \psi(\varphi) = \theta + (i-1)\frac{2\pi}{n} - \sin(\varphi)\tan(\beta), \quad (92)$$

ahol β a csavarvonal menetemelkedési szöge és n az élek száma és i jelenti az aktuális élet. (92)-ot ha behelyettesítjük a csavarvonal paraméteres egyenletébe, akkor a:

$$\begin{aligned} x &= R(\varphi) \cos \theta_i(\varphi) \\ y &= R(\varphi) \sin \theta_i(\varphi), \end{aligned} \quad (93)$$

alakú egyenletet kapjuk, ahol $R(\varphi) = \sqrt{R - (R - R \cos(\varphi))^2}$ a félgömbön és $R(\varphi) = R$ a hengerfelületen. Nagyon szemléletesen lehet (93)-et a Matlab szoftvercsomag segítségével vizualizálni. (lásd. 37.ábra jobb oldala)



37. ábra: a bal oldali ábra a modellben használt szögek rendszerét mutatja. P pont egy tetszőleges pontot jelöl az i . élen. Az ábra azt mutatja, hogyan lehet felírni P helyzetét a szerszám elfordulás szöge ϑ és a φ szögek segítségével. A jobb oldali ábra Matlab rendszerrel készült szerszámmodell négy éllel, jól láthatóak az állandó menetemelkedésű csavarvonalak a félgömb és a hengerfelületen egyaránt.

Martelloti szerint [58] a forgácsvastagság jól közelíthető (87) szerint is, de általános esetben a [68] által javasolt általános képletet érdemes használni:

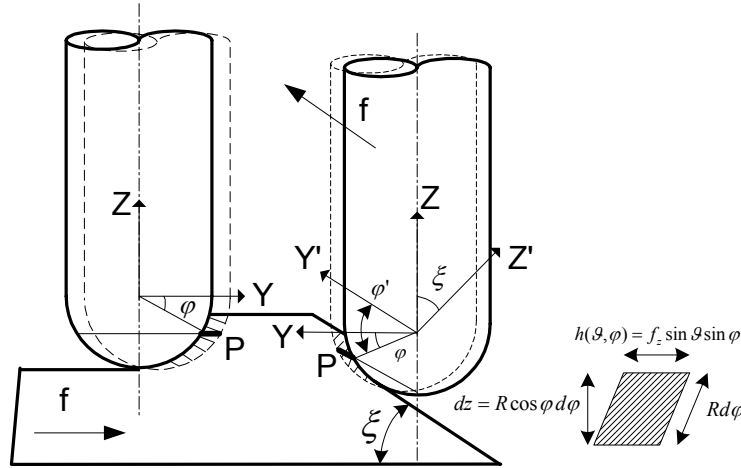
$$h(\vartheta, \varphi) = f_z \sin \vartheta \sin \varphi. \quad (94)$$

38. ábrán látható, hogy dz kifejezhető a maró sugarával és φ szög segítségével a következőképpen:

$$dz = R \cos \varphi d\varphi. \quad (95)$$

Amennyiben a szerszám forgástengelye és az előtolás iránya nem merőleges, például valódi 3 tengelyes megmunkálások esetén, akkor az XYZ koordináta rendszer elfordul ξ szöggel X tengely körül. Az elemi él (P) helyzete:

$$P = (R \cos \vartheta \cos \varphi, R \cos \vartheta \sin \varphi, -R \sin \varphi). \quad (96)$$



38. ábra: amennyiben a szerszám forgástengelye és az előtolás iránya nem merőleges, P pont helyzete az új $Y'Z'$ koordináta-rendszerben φ' lesz, tehát $XYZ \rightarrow X'Y'Z'$ koordináta transzformáció szükséges, amely jelen esetben X tengely körüli ξ szöggel való elforgatással valósítható meg.

Az X tengely körüli ξ szöggel való elforgatás mátrix alakban:

$$R_{Xrot} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \xi & -\sin \xi \\ 0 & \sin \xi & \cos \xi \end{bmatrix}. \quad (97)$$

Ezáltal P pont az $X'Y'Z'$ koordináta-rendszerben:

$$\begin{aligned} P &= (R \cos \vartheta \cos \varphi, R \cos \vartheta \sin \varphi \cos \xi + R \sin \varphi \sin \xi, R \cos \vartheta \sin \varphi \sin \xi - R \sin \varphi \cos \xi) = \\ &= (R \cos \vartheta' \cos \varphi', R \cos \vartheta' \sin \varphi', -R \sin \varphi'). \end{aligned} \quad (98)$$

Ez alapján könnyen kiszámíthatjuk az összefüggéseket a vesszős és vesszőtlen szögek között:

$$\varphi' = -\sin^{-1}(\cos \vartheta \sin \varphi \sin \xi - \sin \varphi \cos \xi), \quad (99)$$

$$\vartheta' = -\cos^{-1}\left(\frac{\sin \varphi \cos \vartheta \cos \xi - \sin \varphi \sin \xi}{\sin \varphi'}\right). \quad (100)$$

A pillanatnyi forgácsvastagság (88) alapján számolható. A tangenciális és radiális erő a következőképpen alakul (90) alapján:

$$\begin{aligned} dF_T &= K_T(\varphi) [f_z \bullet N(\mathcal{G}'(\mathcal{G}, \varphi))]^{m_T} R \cos \varphi' d\varphi', \\ dF_R &= K_R(\varphi) [f_z \bullet N(\mathcal{G}'(\mathcal{G}, \varphi))]^{m_R} R \cos \varphi' d\varphi'. \end{aligned} \quad (101)$$

A két erőkomponens alapján a dF_x és dF_z a következő képlet alapján számítható:

$$\begin{bmatrix} dF_x \\ dF_y \\ dF_z \end{bmatrix} = R_z(\mathcal{G}) \begin{bmatrix} dF_t \\ dF_r \\ dF_a \end{bmatrix}, \quad (102)$$

ahol:

$$R_z(\mathcal{G}) = - \begin{bmatrix} \cos \mathcal{G} & \sin \mathcal{G} & 0 \\ -\sin \mathcal{G} & \cos \mathcal{G} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (103)$$

(102) kifejtve és összegezve a teljes vágóél mentén:

$$\begin{aligned} F_x(\mathcal{G}) &= \int \rho R \{ -K_T(\varphi) [f_z \bullet N(\mathcal{G}'(\mathcal{G}, \varphi))]^{m_T} \cos \mathcal{G} - K_R(\varphi) [f_z \bullet N(\mathcal{G}'(\mathcal{G}, \varphi))]^{m_R} \sin \mathcal{G} \} \cos \varphi' d\varphi' \\ F_y(\mathcal{G}) &= \int \rho R \{ K_T(\varphi) [f_z \bullet N(\mathcal{G}'(\mathcal{G}, \varphi))]^{m_T} \sin \mathcal{G} - K_R(\varphi) [f_z \bullet N(\mathcal{G}'(\mathcal{G}, \varphi))]^{m_R} \cos \mathcal{G} \} \cos \varphi' d\varphi' \end{aligned}$$

$$\rho = \begin{cases} 1 & \text{ha részt vesz a forgácsképzésben} \\ 0 & \text{nem} \end{cases}. \quad (104)$$

A háromdimenziós érintkezési terület (contact area), az OpenGL stencil buffere segítségével a szerszám mozgása közben bármilyen helyzetben előállítható, amelyeket diszkrét pontok halmazaként a szerszám síkjára vetítünk. Hasonlóképpen a szerszám él elemek is ugyanerre a síkra vetíthetők. A ρ értéke 0 vagy 1 attól függően, hogy az adott szerszámelem részt vesz-e a forgácsolásban vagy nem (azaz egybeesik-e a szerszám síkján az érintkezési terület egy pontjával, vagy nem). Az adott éldarab sugara a következő képletből számítható $R(\varphi) = \sqrt{R - (R - R \cos(\varphi))^2}$, az X és Y irányú helyzete pedig a csavarvonal egyenlete alapján adódik (93). Ha az élelem kiszámított (X, Y) pozíciója részt vesz a forgácsképzésben, akkor (101) alapján az elemi forgácsolási erő számítható, és numerikus integrálás segítségével a pillanatnyi erő (104) alapján adódik. A pillanatnyi erő maximumát

nehéz előre meghatározni, ezért általában a numerikus integrálást több forgásszögre is el kell végezni.

5.2.5 Az empirikus konstansok meghatározása

Valójában az előző fejezetben említett konstansokat nem könnyű meghatározni mert ezek számos változó függvényei, mint például a munkadarab anyaga, hűtés, a szerszám hátszöge, forgácsvastagság, szerszámhőmérséklet. Spence és Altintas [referencia] mutattak be egy módszert a konstansok meghatározására, átlag-forgácsolási erő mérésével sok egyszerű geometriájú próbamegmunkáláson keresztül. Főként gömbvégű szerszámok mérésével foglalkozott Feng és Menq [64], akik gyakorlati módszert dolgoztak ki a $K_T(\varphi)$, $K_R(\varphi)$ és m_R , m_T meghatározására.

5.3 A biztonságos megmunkálási feltételek ellenőrzése

A továbbiakban azt fogjuk vizsgálni, hogyan lehet az előzőekben bemutatott modellek segítségével a biztonságos forgácsolási feltételeket ellenőrizni. Számos feltételt lehet figyelembe venni, de ezek közül jelen dolgozatban a fogra ható maximális feszültséget és a szerszám elhajlást fogjuk részletezni.

5.3.1 A fogakban ébredő feszültség

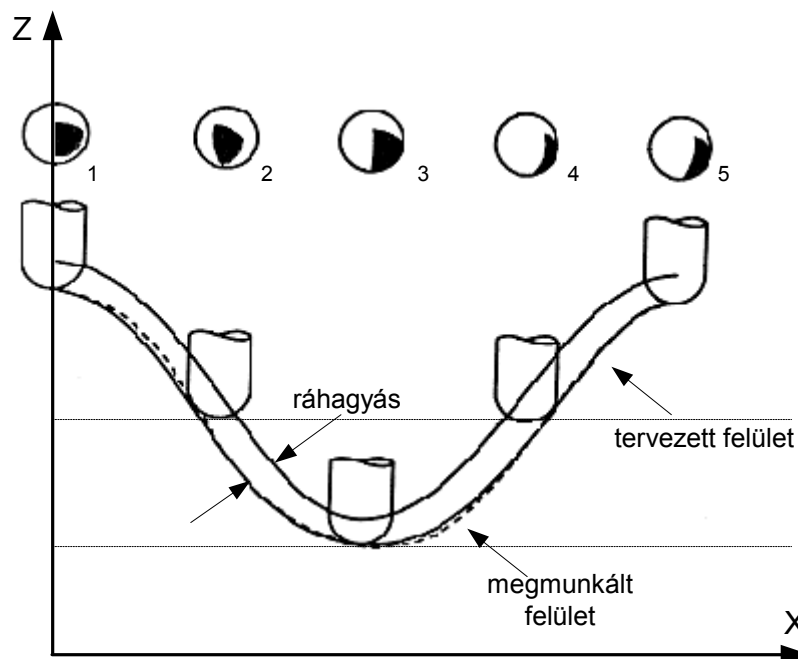
Lapka vagy éltörés következik be, ha a fogra ható erő meghaladja az anyagára jellemző szakítószilárdságot. Mivel a lapkában ébredő feszültség egyenesen arányos a forgácsvastagsággal, ezért meg kell keresnünk azt a helyet az érintkezési tartományon belül, ahol ez maximális. Ebben az esetben nincs szükség a forgácsolási erő kiszámítására különböző forgásszögeknél, tehát a számítási idő jelentősen csökken. A bemutatott modell szerint az algoritmus a következő:

1. a stencil buffer által jelzett pontokban ki kell számítani a forgácsvastagságot a (88) alapján
2. a kiszámított h_{max} maximális forgácsvastagságot össze kell hasonlítani a megengedhető forgácsvastagsággal h_{meg}
3. ha $h_{max} > h_{meg}$ akkor csökkenteni kell a fogankénti előtolást addig, amíg a feltétel nem teljesül

Ez az egyszerű számítás alkalmazható például nagyolás esetén. Simítás esetén gyakorlatban a szerszámelhajlást célszerű vizsgálni.

5.3.2 Szerszámelhajlás miatt keletkező felületi hiba

Szerszámelhajlás ellenőrzése kulcsfontosságú a marás-szimulátorokban, a 39. ábra mutatja, hogy az előtolás z komponense milyen hatással van a megmunkált felület minőségére abban az esetben, ha állandó előtolással és főorsó sebességgel történik a megmunkálás. Tehát a beállított forgácsolási sebességet és a fogankénti előtolást táblázatok alapján állítjuk be. Jelenleg a CAM rendszerek nagy része ezt a hagyományos módszert alkalmazza a szerszámpálya generálásakor.



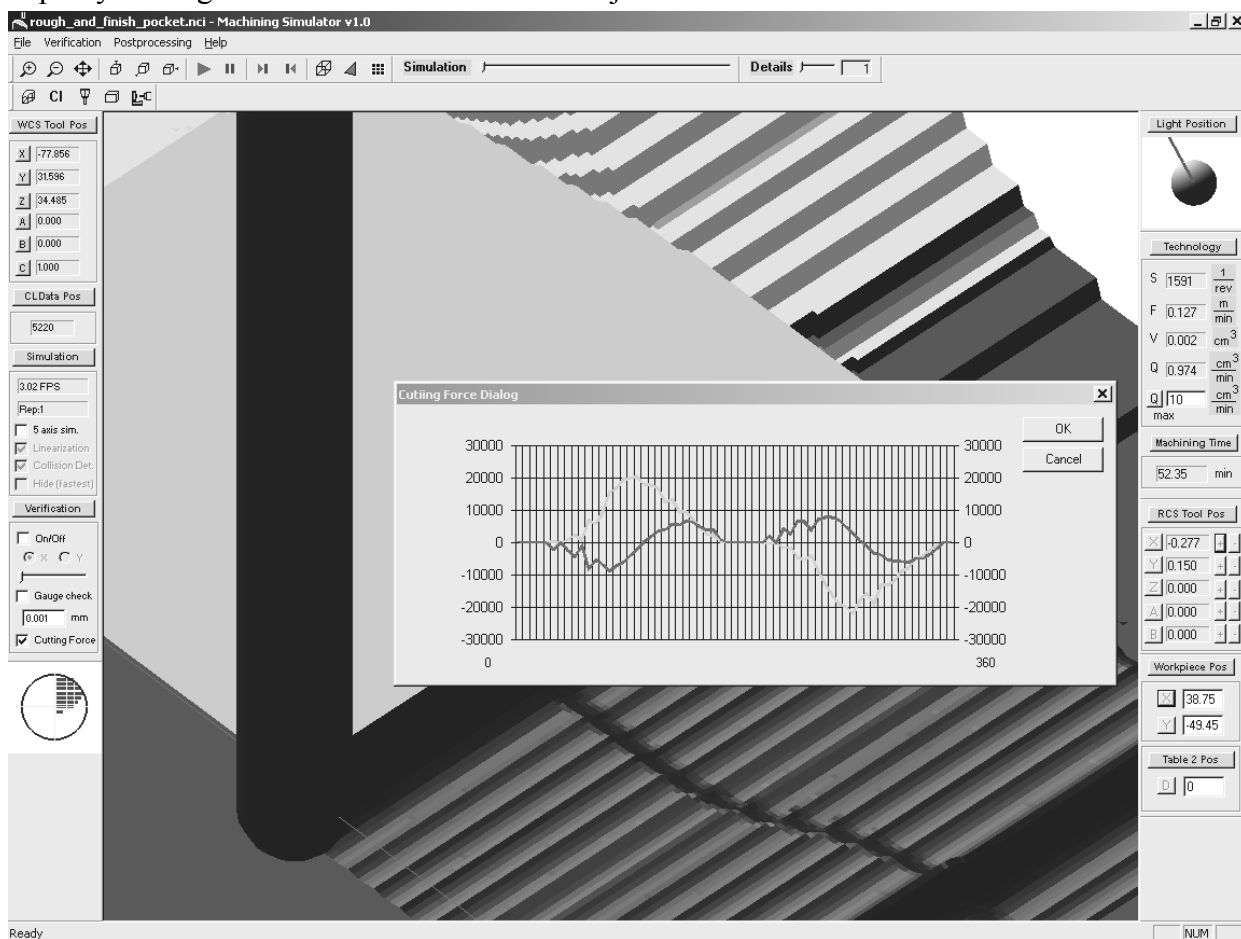
39. ábra: szerszámelhajlásból adódó felületi hiba 5 különböző szerszámhelyzetben [26].

Az ábra alapján jól követhető, ha a szerszám -z irányba mozog, akkor az elhajlása következtében a felületen ráhagyást hagy (lásd az ábra bal oldala), ha pedig emelkedik, akkor alámetszést okoz. A szimulátorban ezt a hatást a következőképpen lehet számolni: a (104) képlet szerint meg kell keresni, milyen ϑ forgásszög mellett adódik abszolút értékben a legnagyobb erő, és ezt az erőértéket kell alkalmazni a kihajlás (42) képletben:

$$\delta_x = \frac{F_x L^3}{3EI} \quad \delta_y = \frac{F_y L^3}{3EI}. \quad (105)$$

5.3.3 A módszer alkalmazása a saját fejlesztésű megmunkálás-szimulátorban

Már többször is említettem, hogy ebben a fejezetben leírt módszerek jól alkalmazhatóak a saját fejlesztésű megmunkálás-szimulátorban, amelynek bemutatásával a következő fejezet foglalkozik. Gyakorlati méréseket nem végeztem, a megvalósított eljárás demonstrációs célt szolgál. Az empirikus konstansokat tetszőleges értékekre állítottam be. A 40. ábrán látható képernyő a megmunkáláskor ébredő erőt mutatja az



40. ábra: a fellépő erők analitikus modellezése a p-bufferes szimulátorban. A megjelenő ablak a radiális és tangenciális erőkomponensek nagyságát mutatja 0-360°-os teljes szerszám körbefordulás esetén. A szerszám pillanatnyi érintkezési területét a bal oldali keretben lévő kör sötétített része jelzi.

erő és elfordulási szög koordináta rendszerben. A p-bufferben az erőszámítást a szerszám egyes mozzanatain belül, raszterenként kell elvégezni. Akkor kell csökkenteni az előtolást, ha az erő valamelyik komponense, egy előre megadott határértéket túllép.

Továbbfejlesztésként, és a módszer használhatóságának igazolására gyakorlati méréseket is kell végezni, amelyekkel meg lehet határozni az empirikus konstansokat.

6 A prototípus megmunkálás-szimulátor szoftver bemutatása

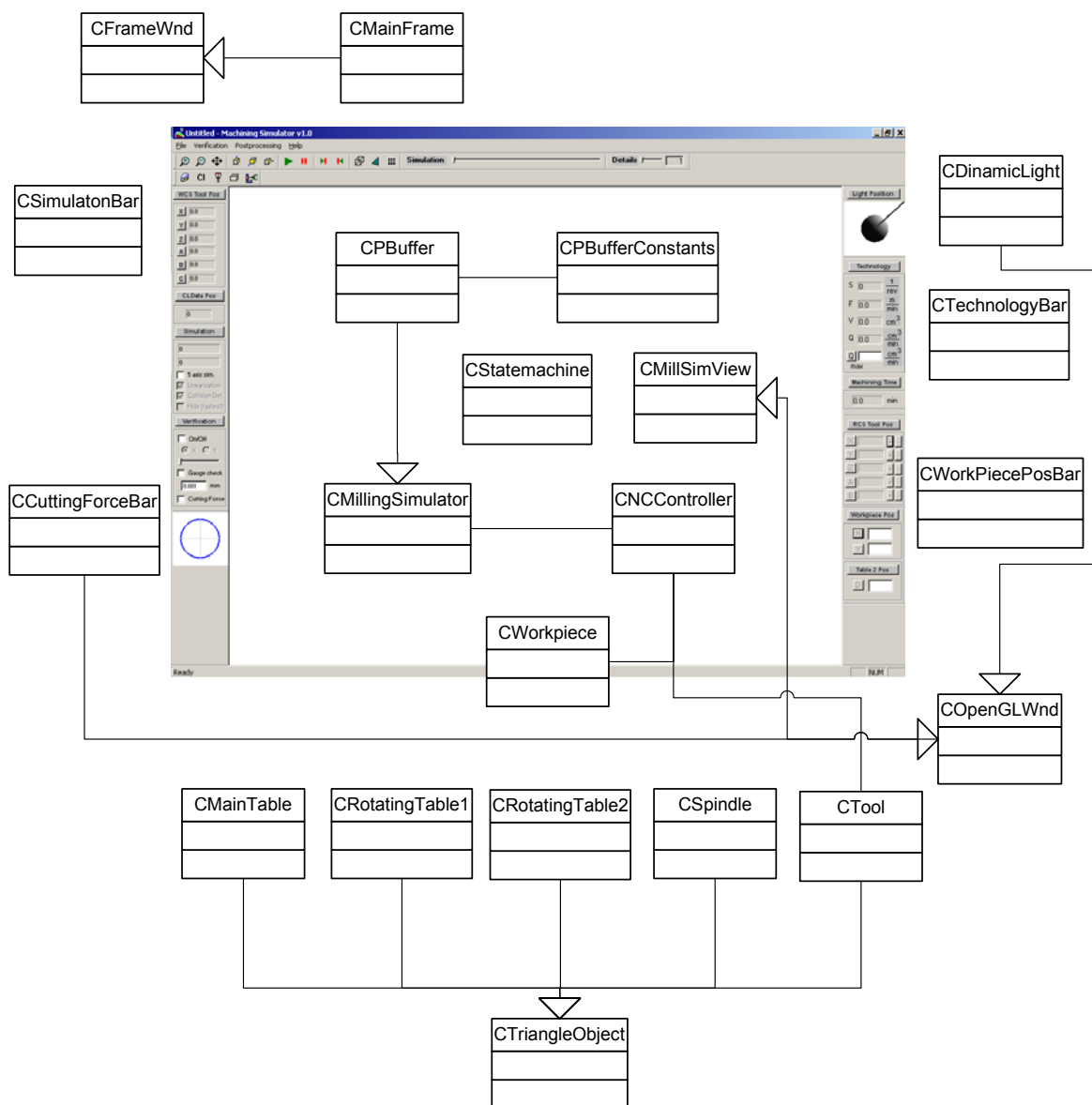
Az eddigiekben, elméleti oldalról mutattam be az új p-buffer eljárást, és a modellezés fontosabb kérdéseit. Ebben a fejezetben a módszer köré épített prototípus szimulátor program, és a legfontosabb gyakorlati implementációs kérdések kerülnek bemutatásra.

Manapság az objektum orientált programozás paradigmája, számos előnye miatt egyeduralkodó, így nem volt kérdés számomra valamilyen OOP környezet alkalmazása. A Microsoft Visual C++ rendszer használata mellett döntöttem. Alternatívaként szóba került a Microsoft Visual Basic felhasználása is, de a jelen feladat szempontjából ez azért nem előnyös, mert a VB automatikus memóriamenedzsmentje tapasztalataim szerint nem hatékony. A jelen példaalkalmazásnál pedig stratégiai jelentőségű a nagy memóriaterületek gyors és rugalmas kezelése. A felhasználói felület MFC osztályok felhasználásával készült. A szimulátor implementálásához érdemes volt olyan eszközöket is alkalmazni, amelyek megkönnyítik a nagyméretű tömbök és mátrixok kezelését. Erre a feladatra az STL osztálykönyvtárat tartom a legalkalmasabbnak [40], amely szabványosított generikus osztályokat tartalmaz. A 41. ábra a szimulátor statikus osztály diagrammját mutatja, a főablak köré csoportosítva. Nem tüntettem fel a szabványos alaposztály készletekből (STL, MFC) történő öröklődéseket, az attribútumokat és az operációkat, a koncepció egyszerűbb megértése, átláthatósága érdekében.

A *CNCController* a szerszámgép vezérlést modellelzi, tárolja az aktuális szerszámot, szerszámpozíciót, a soron következő CLData pozíciót, felismeri a 3 és 5 tengelyes mondatokat, transzformál, és linearizál ha szükséges, e mellett elkészíti a posztprocesszált NC kimenetet is. Szoros kapcsolatban van a *CMillingSimulator* osztállyal, ami a geometriai szimulációt, forgácsolási erő, és forgácsleválasztási intenzitás-számítást végez. A *CPBuffer* osztályban valósítottam meg az előző részben bemutatott új szimulációs módszert, amely szorosan együttműködik az OpenGL API-val. A *CStateMachine*, egy olyan állapotgép, amely a szimulátor egyes állapotait tárolja a felhasználó szemszögéből. Az egyes vezérlőelemek engedélyezett vagy nem-engedélyezett állapotait szabályozza a benne létrehozott kezelő függvények és változók segítségével.

A *CSimulationBar*, *CCuttingForceBar*, *CWorkpiecePosBar*, *CTechnologyBar*, *CDinamicLight* osztályok objektumainak nézetei, a felhasználó által állítható vezérlő elemek egész sorát tartalmazzák, melyek állapotait az állapotgép osztály felügyeli. Számos dialógus ablakból származó osztály öröklődik a *COpenGLWnd* osztályból egyaránt, mivel ez a bázisosztály nagymértékben általánosítja az OpenGL közös megjelenítő metódusait. A *CMainTable*, *CRotatingTable1*, *CRotatingTable2*, *CSpindle*, *CTool* osztályok egyaránt a *CTriangleObject* bázisosztályból származnak. Ez az öt osztály reprezentálja a két körasztalos megmunkáló központ geometriáját. A *CTriangleObject* bázisosztály beolvassa a Pro/Engineer által generált geometriai fájlokat, ha egy modell több alkotórészből áll, például állórészből és forgórészből, akkor az egyes részeket dinamikusan, külön tárolja. Ez az öt osztály képes tárolni

a modellek helyzetét, inicializálja az ütközés felügyeletért felelős *Solid 2.0* statikus könyvtárát, és nem utolsósorban, a munkatérben megjeleníti a tárolt modelleket.



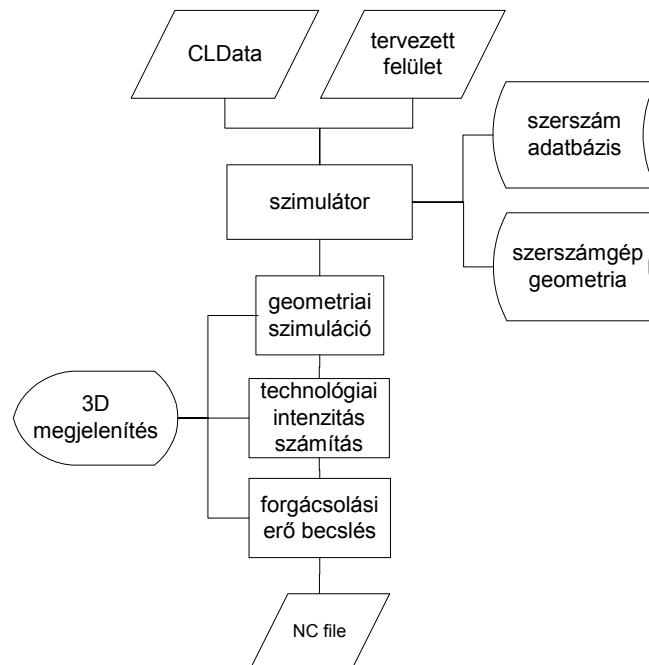
41. ábra: az ábra a szimulátor szoftver főbb osztályait mutatja, a felhasználói interfész köré csoportosítva. Nem tüntettem fel a szabványos alapsztály készletekből (STL, MFC) történő örökléseket, az attribútumokat és az operációkat, a koncepció egyszerűbb megértése érdekében.

6.1 A szimulátor főbb funkciói

A konkrét funkciók bemutatása előtt meg kell említenem azt, hogy a szoftver prototípus kifejlesztésekor nem volt céлом jobb és robusztusabb szimulátort létrehozni, mint a kereskedelmi forgalomban kaphatóak. Ilyen célok megvalósításához egymagam nem vagyok

elég, mivel a mai kereskedelmi programokat szoftverfejlesztő csoportok fejlesztik, több tíz programozó év munkával. A szimulátorom fő feladata az, hogy bizonyítsa, a p-buffer eljárás iparban való alkalmazhatóságát. A következő öt fontos funkció implementálását tartottam fontosnak: a felület pontosságának számítása; gyors, nézőpont független grafikus megjelenítés; a gépgeometria modellezése; a forgácsleválasztási intenzitás, és a forgácsolási erő számítására. Ennek a néhány jellemzőnek a megvalósításához is tetemes programozói munkára volt szükség (a szoftver prototípus forráskódja több mint tízezer programsort tartalmaz).

A 42. ábrán a szimulátor funkcionális modellje látható. A rendszer bemente, a szerszámpálya generátor CAM által létrehozott CLData fájl és a CAD által tervezett felület modell. Bemenetként funkcionál a belső szerszám adatbázis, amely a legalapvetőbb információkat tartalmazza a szerszámokról (átmérő, hossz, típus, index, programozott pont helye). A szerszám adatbázis rekordjai a szimulátoron belül interaktívan vagy egy szövegszerkesztővel módosíthatók. Az 5 tengelyes szerszámgép geometria kötött, ez a szoftververzió csak ezt a két körasztalos konfigurációt tudja betölteni. Más szerszámgép kialakítások beillesztésére is megvan az elvi és gyakorlati lehetőség, mivel a felhasznált algoritmusok teljesen általánosak.



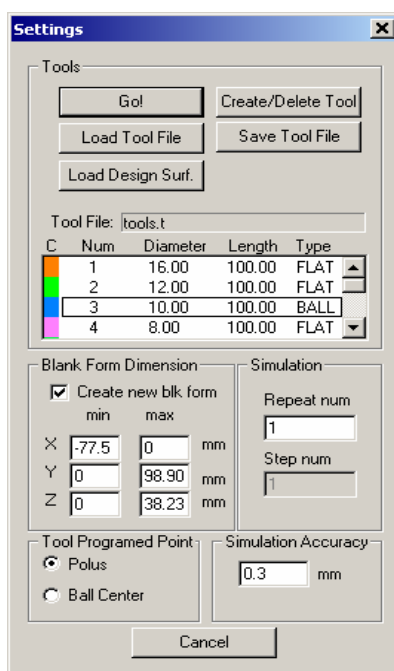
42. ábra: a szimulátor funkcionális modellje. Bemenetként a CLData és a tervezett felület használható. A rendszer belső adatbázisa, a szerszám adatbázis és a szerszámgép konfiguráció geometria. Kimenetként a háromdimenziós megjelenítés, és az NC fájl szolgál.

A szimulátor három különböző funkciót lát el, ezek közül első, a geometriai szimuláció, amelyről az előző fejezetekben részletesen beszámoltam. A további két funkciót a további fejezetekben fogom tárgyalni. A szimulátor program egyik kimenete a monitor képernyője,

amelyen szemügyre vehetjük a megmunkálást animáció segítségével, és megfigyelhetjük a kialakult geometriai hibákat. Másik kimenete pedig maga az NC fájl, amely szabványos ISO formátumban tartalmazza a szerszámpályákat. Véleményem szerint ezek a legalapvetőbb funkciók, amelyeket meg kellett valósítanom ahhoz, hogy kellő mértékben bizonyítható legyen a p-buffer módszer használhatósága. A szimulátor mindhárom fő funkciója felhasználja a p-buffer által szolgáltatott részeredményeket.

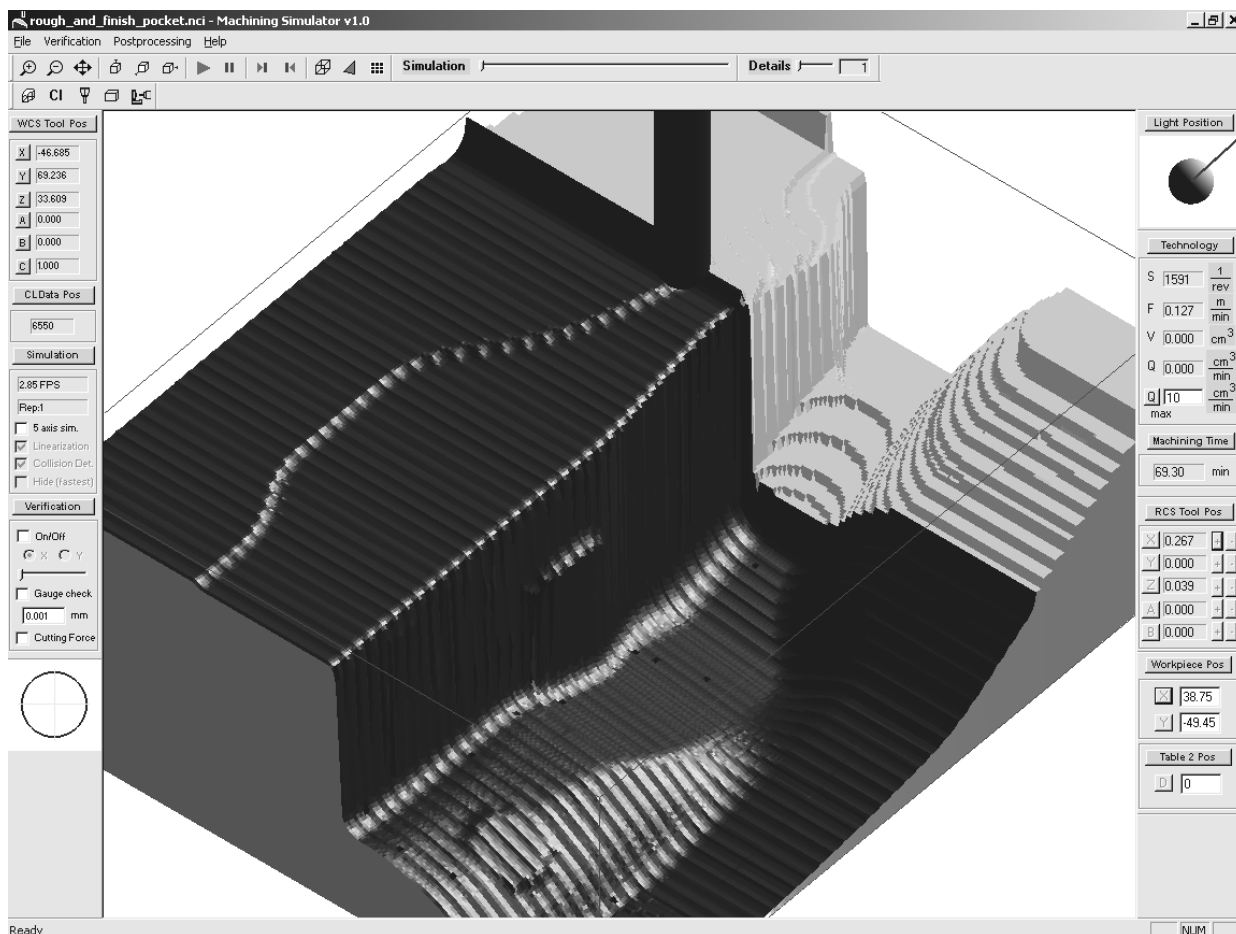
6.2 A funkciók bemutatása példák alapján

A szimulátor teszteléséhez két szerszámpályát használtam fel az egyszerűbb néhány soros tesztfájlok után. Első egy háromtengelyes, a másik egy öttengelyes szerszámpálya volt. Mindkettő nagyolás, félsimítás és simítás műveletekből épül fel különböző méretű és típusú szerszámok felhasználásával. Mindkét példafájl megtalálható a MasterCAM szoftver példatárában. A háromtengelyes szimuláció a következőképpen történik: első lépésben bemenetként meg kell adni a szerszámpályát és a kész darab geometriát. A 43. ábra mutatja a lehetséges beállítási lehetőségeket. Az alapértelmezett szerszámfájl a tools.t, amelynek az információi a listadobozban láthatóak. A nyersdarab méretét a szoftver automatikusan állítja be, a CLData alapján.

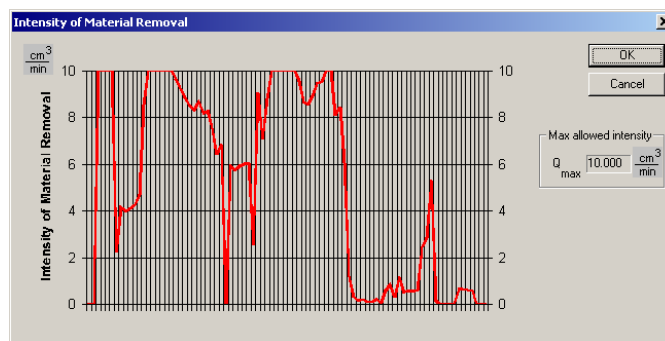


43. ábra: a szerszámpálya betöltése után megjelenő dialógusablak, amely a tools.t fájlban tárolt szerszámok információit jeleníti meg, beállítható, módosítható a nyersdarab mérete, néhány szimulációs tulajdonság

A 'Go' gomb megnyomása után a nyersdarab geometria feldolgozásra kerül, majd a szerszámpálya szimuláció megkezdhető. A geometriai szimuláció egy köztes fázisa a 44. ábrán látható, az alsó részben már a kész simított felület figyelhető meg, a jobb felső részben az előző félsimító művelettel kialakított felület eltérő színnel jelenik meg. A szimulátor méri a leválasztott anyagterfogatot, ennek intenzitását és a megmunkálási időt. Beállítható egy Q_{max} maximális intenzitás, amelynek túllépése esetén a szoftver úgy csökkenti az előtolást, hogy az ezzel kialakuló új intenzitásérték Q_{max} legyen.

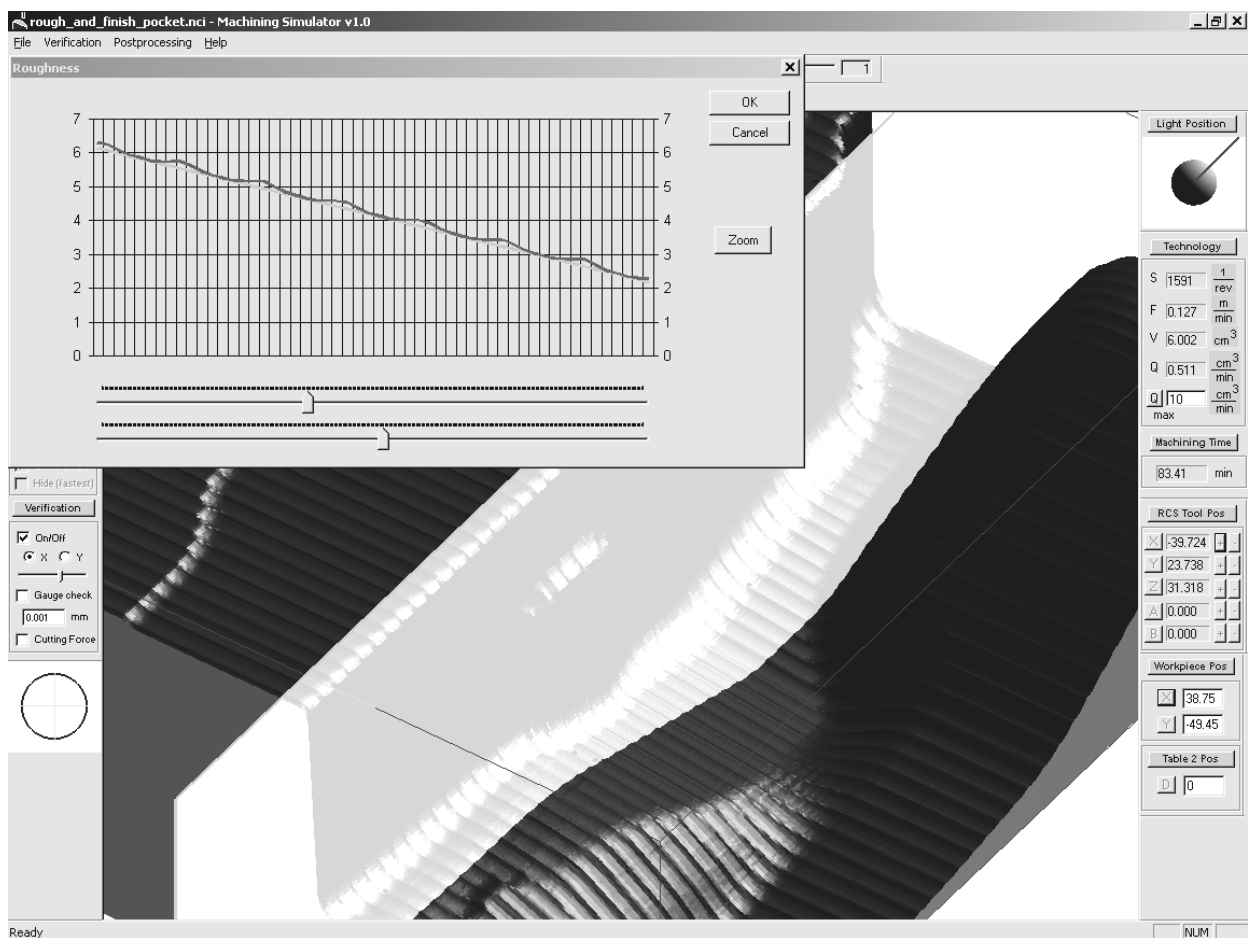


44. ábra: a geometriai szimuláció egy köztes fázisa: az alsó részben már a kész simított felület figyelhető meg, a jobb felső részben az előző félsimító művelettel kialakított felület eltérő színnel jelenik meg.



45. ábra: az anyagleválasztás intenzitásának mérése. $Q_{max} = 10 \text{ cm}^3/\text{min}$ határintenzitás beállítása után a rendszer szabályozza, és nem engedi túllépni ezt a határt. Túllépés esetén úgy csökkenti az előtolást, hogy az intenzitás éppen a megengedett legnagyobb legyen (Q_{max}).

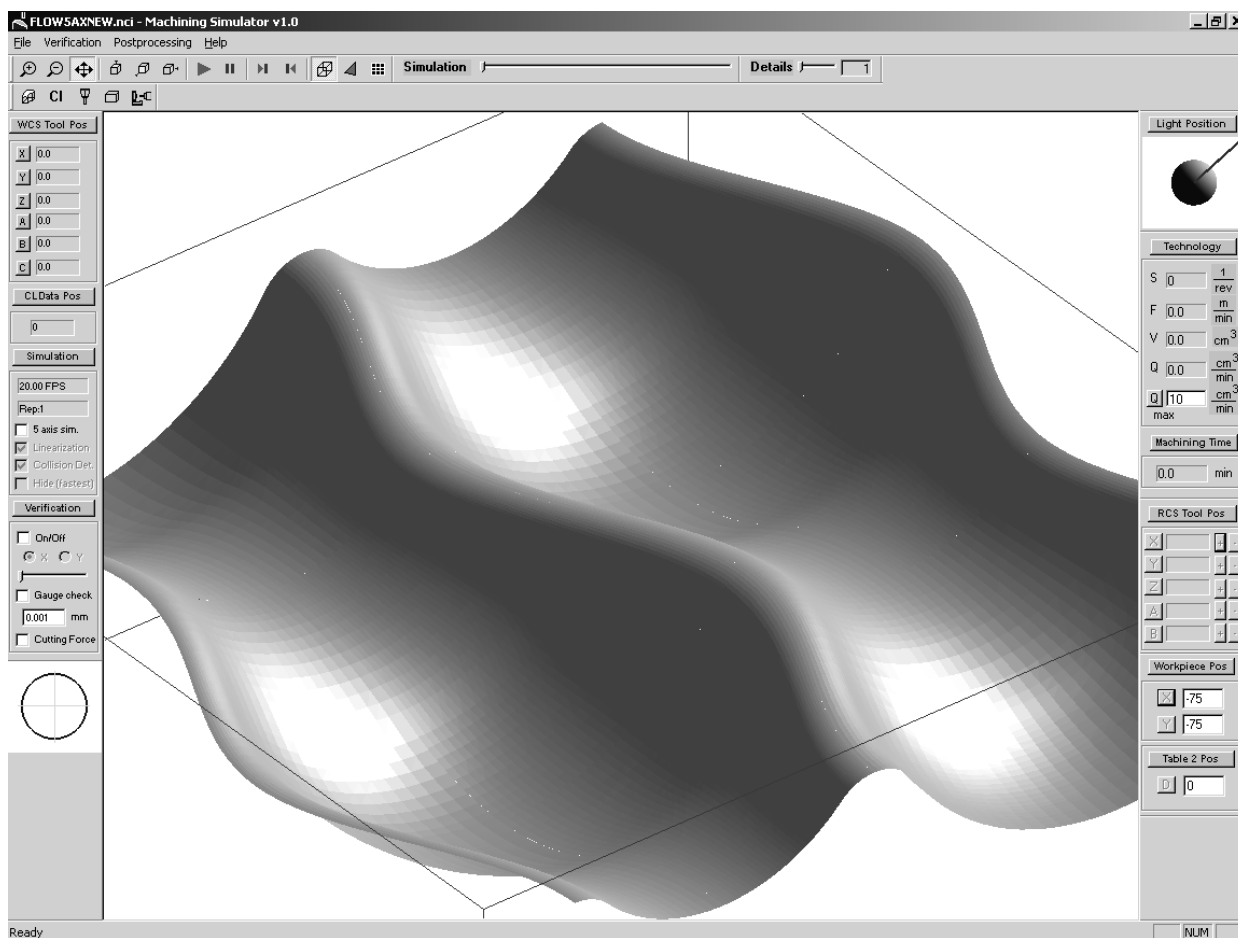
A 45. ábrán látható a szoftver anyagleválasztási intenzitást figyelő ablaka. Az ablak, egy nem modális dialógusablak, a megmunkálás-szimuláció közben mindig láthatóvá tehető. Az intenzitásérték kirajzolását az *MSChart* kontroll végzi, amely továbbfejlesztés esetén kicserélhető más diagramrajzoló eszközre is.



46. ábra: a kész felület megmunkálási pontosságának vizsgálata félig átlátszó síkfelület mentén. A kész felület és a tervezett felület eltérése nyomon követhető a megjelenő ablakban. A sík tetszőlegesen mozgatható, és *zoom* gomb segítségével a sík mentén nagyítani és kicsinyíteni is lehet a vizsgálandó intervallumot.

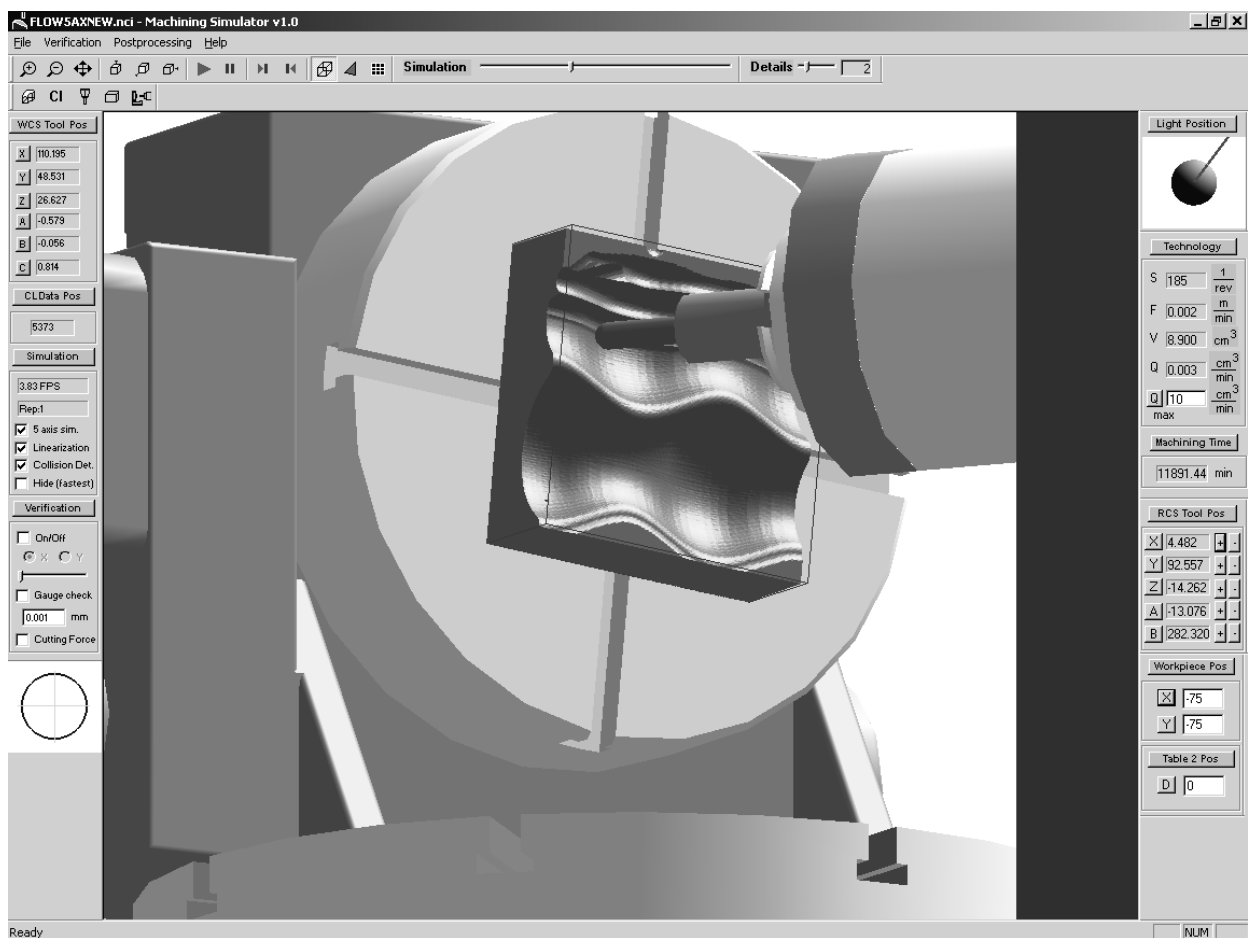
A 46. ábra a kész felület megmunkálási pontosságának a legegyszerűbb vizsgálati módszerét mutatja a szimulátorban. A megjelenő dialógusablakban a kész felület és a tervezett felület egy tetszőleges sík mentén létrejövő metszete figyelhető meg. A metszet síkjában nagyítani, kicsinyíteni lehet a vizsgált tartományt. Ezzel a módszerrel csak nagy eltérések figyelhetők meg teljes biztonsággal. Ennél kisebb eltérések vizsgálata más grafikus módszerrel történik. A bal oldali keretben egy 'Gauge check' jelölőnégyzet segítségével bekapcsolható az alámetszés ellenőrzés. Ennek bekapcsolása után, a megmunkált felületen piros színnel jelennek meg azok a felületrészek, ahol az előírtnál nagyobb alámetszés keletkezett.

Az öttengelyes marás-szimulációjának lépései annyiban térnek el a háromtengelyestől, hogy munkatér szimuláció is kiválasztható. A 47. ábrán az öttengelyes példában felhasznált szabadformájú felületet látható.



47. ábra: az öttengelyes példában felhasznált szabadformájú felületet ábrázolja.

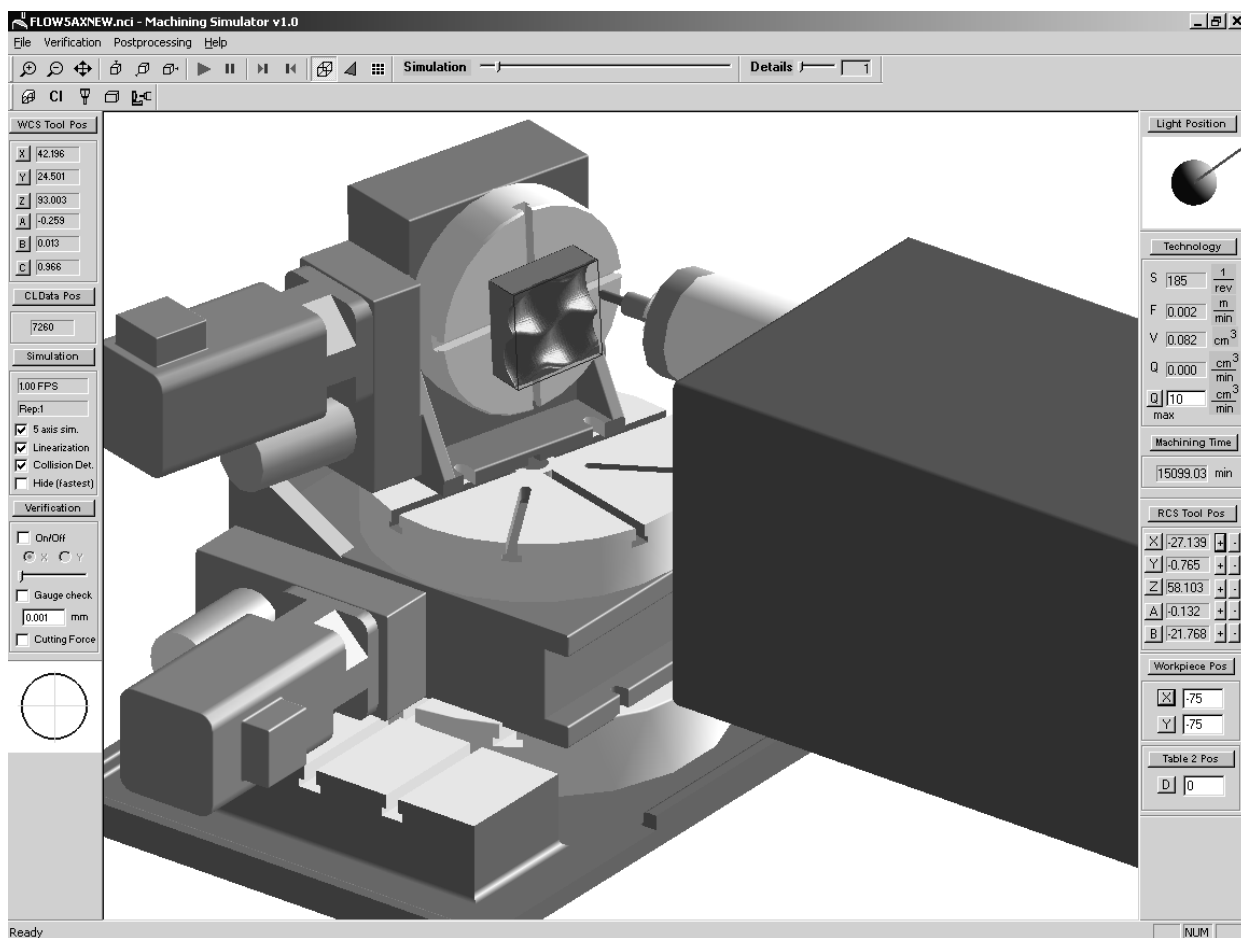
A szoftver bemenetét öttengelyes szimuláció esetén is ugyanazok az információk képezik, mint hagyományos háromtengelyes szerszámpályák esetében. A különbség csak annyi, hogy amennyiben munkatér szimuláció is szükséges, akkor a gépkonfiguráció geometriát is be kell tölteni. A szimuláció megkezdése előtt fel kell venni a nullpontot. A jobb oldali keret 'RCS Tool Pos.' (RCS - Relative Coordinata System) '+' és '-' nyomógombjaival az öt tengely tetszőleges pozícióba mozgatható, az ütközésvizsgáló rendszer csak addig engedi a mozgatót, amíg a szerszám meg nem érinti a munkadarabot, vagy a körasztalt. A relatív koordináták nullázása, az 'X', 'Y', 'Z', 'A', 'B' nyomógombok megnyomása után megy végbe. A transzformációs (linearizációs) eljárás is relatív koordinátákat ad eredményül, melyek posztprocesszálás után a valóságos szerszámgép vezérlésében is végrehajthatóak. Esetleges nem kívánt ütközések esetén a munkarab pozíciója állítható a jobb oldali keret 'Workpiece pos.' feliratú részében. Az egyes és kettes körasztal egymáshoz viszonyított helyzete is ugyanitt beállítható. Az öttengelyes munkatérbeli anyagleválasztás szimuláció egy köztes fázisa a 48. ábrán figyelhető meg.



48. ábra: az öttengelyes munkatérbeli anyagleválasztás szimuláció egy köztes fázisa figyelhető meg az ábrán. A munkadarab alsó része már készen van, az 5373 szerszámpozíció (mozzanat) szimulálása után. A megmunkálás tetszőleges nézőpontból megfigyelhető.

A szimulátor nagy előnye az, hogy a valós gépkonfigurációt és az anyagleválasztást együttesen képes szimulálni, és a folyamat animáció segítségével, jó minőségben végigkövethető. Az animáció sebessége egy közepes gépen 15-20 képkocka másodpercenként. Jobb grafikus kártya alkalmazásával ez a sebesség növelhető. A következő ábrán a kész munkadarabot figyelhetjük meg egy olyan nézetben, amelyben a teljes gép látható. A munkadarab rögzítő és felfogószerkezete nincs megjelenítve, illetve modellezve a munkatérben. A felfogószerkezetek általában egyedi kialakításúak, továbbfejlesztésként érdemes egy interaktív eljárást kidolgozni a felfogások kiválasztásához, elhelyezéséhez. A szoftverben használt jól általánosítható gépgeometria modellezési módszer lehetővé teszi az ilyen irányú továbbfejlesztéseket is.

A 49. ábra a kész munkadarabot mutatja a teljes szerszám pálya szimuláció után. (több mint 7000 szerszámhelyzet után)



49. ábra: a kész munkadarab a teljes szerszámpálya szimuláció után, amely 7360 szerszámhelyzetet (szerszámmozzanatot) tartalmaz.

6.3 A szimulátor sebességének összehasonlítása

A p-bufferrel segített szimuláció számos előnnyel rendelkezik. A legfontosabb ezek közül a rendkívüli gyorsaság. A teszteredmények azt bizonyítják, hogy a sebessége szinte kizárólag a grafikus kártya sebességétől függ, a processzor sebessége kevésbé vagy egyáltalán nem befolyásolja. További fontos előnye az egyszerű implementálhatóság (lásd 13. ábra). A következő táblázat ugyanazon processzorral és különböző videokártyák alkalmazása esetén mutatja be az egy másodperc alatt szimulált szerszámhelyzetek számát (természetesen azonos CLData teszt-szerszámpályát alkalmazva).

	processzor	videokártya	sebesség (szerszámhelyzet/ per secundum)	Szimulációs módszer
1.	Intel Pentium II 333 Mhz	tetszőleges	70	gamma buffer
2.	Intel Pentium III 666 Mhz	tetszőleges	130	gamma buffer
3.	Intel Pentium III 666 Mhz	tetszőleges	100	octal tree
4.	Intel Pentium III 666 Mhz	Matrox G-400	1300	p-buffer
5.	Intel Pentium III 666 Mhz	NVIDIA TNT 2	1300	p-buffer
6.	Intel Pentium III 666 Mhz	NVIDIA GeForce 2 MX 400	10000	p-buffer

50. ábra: a p-bufferes eljárás sebességének összehasonlító eredményei.

A fenti ábra alapján (50. ábra) a következő következtetések vonhatók le:

1. A hagyományos módszerek (vö. 1. 2. 3.) sebessége az alkalmazott videokártyától természetesen nem függ, mivel az nem vesz részt az anyagleválasztás szimulációjában. Kizárólag a processzor sebessége határozza meg a sebességet.
2. Gyorsabb processzor alkalmazása az órajelek arányában növeli a szimuláció sebességét. (vö. 1. 2.)
3. A p-buffer módszer ugyanazon a processzoron, gyorsabb videokártya alkalmazása mellett legalább tízszeres sebességnövekedést okoz. Megjegyzendő az, hogy napjainkban csaknem lehetetlen olyan videokártyát vásárolni, amely nem támogatja a hardveres OpenGL gyorsítást (*NVidia TNT 2* ára tízezer forint alatt van). A 5. 6. pontban tesztként használt kártyák teljesen átlagosnak számítanak, a legújabb *GeForce 4 Ti* kártyával nem volt lehetőségem tesztelni A gyártó tesztjei szerint ez legalább háromszor gyorsabb, mint a *TNT2* modell.

A p-buffer jelentős sebességi előnye, egyrészt abból adódik, hogy a háttérben alkalmazott z-buffer algoritmus fixpontos adattárolás esetén MSI elemekkel hardverben könnyen realizálható, másrészt a legújabb grafikus hardverek képesek a raszterizáció folyamatát párhuzamosítani, (például a GeForce 3 kártya már egyszerre négy pixelt képes párhuzamosan számolni) [33,36]. A kereskedelmi forgalomban kapható szimulátor-programokkal való összehasonlításban is hasonló eredményeket tapasztalhatunk. A különálló termékként forgalmazott szimulátorok fejlesztői arra törekednek, hogy a szoftverük egy-egy nagyobb CAD/CAM gyártó komplex szoftverrendszerébe opcionálisan vagy teljes egészében integrálódjon. Erre a legjobb példa a VeriCut és a Pro/Engineer integrációja. Ha a saját p-bufferes szimulátort és az ismert két nagy CAD/CAM rendszer, a MasterCAM és Pro/Enginner szimulátorait összehasonlítom, akkor azt tapasztalom, hogy a saját módszerem szimulációs sebessége lényegesen gyorsabb. A kereskedelmi forgalomban kapható szoftverek készítői nem közölnek semmilyen információt arról, hogy milyen szimulációs módszert alkalmaztak. Véleményem szerint az “extended-dexel” módszert alkalmazzák mindkét esetben. Jelen disszertáció írása közben találtam egy

nagyon ígéretes megvalósítást: a Northwood Designs Inc, Metcut elnevezésű szerszámpálya-szimulációs szoftverét [41]. Az alkalmazott módszer a tesztjeim és tapasztalatom alapján a gamma-buffer eljárás. Ki kell emelni a munkadarab megjelenítés esztétikai szépségét, amit az OpenGL módszer lehetőségeinek csaknem teljes felhasználásával értek el a fejlesztők.

Általában minden diszkrét módszerről elmondható az a hátrányos tulajdonság, hogy 50-100 Mbyte memória is szükséges lehet egy szerszámpálya kellő pontosságú szimulációjához.

6.4 A szoftver jelenlegi változata és továbbfejlesztési lehetőségek

A fejezetben bemutatott szoftver jelenlegi változata dedikált környezetben fut automatikus installálási lehetőség nélkül. Az első lényeges továbbfejlesztési irány egy automatikus installer kifejlesztése és tesztelése lesz különböző operációs rendszereken. A bemutatott szoftver jelenleg is fejlesztés alatt áll. A jövőben szeretném újabb gépkonfigurációkkal kiegészíteni, szerszámadatbázis ODBC kapcsolatot létrehozni. A tanszék által oktatott tárgyak keretében a műszaki-informatikus hallgatóknak demonstratív céllal már többször bemutatásra került. A török METU (Middle East Technical University) egyetemmel közös kutatási projekt keretében is alkalmazásra kerül majd, mivel a török partner élénk érdeklődést mutatott a szimulátor iránt és felajánlotta a közös továbbfejlesztés lehetőségét is.

7 Új tudományos eredmények

Az intelligens alkatrészgyártás egyik legfontosabb törekvése az, hogy a szerszámgépek működése közben minimálisra csökkentsék az operátori felügyelet szintjét. Ennek eléréséhez az NC program szimulációs és ellenőrző szoftverek felhasználása alapvető fontosságú. Megvizsgálva a különböző ismert módszereket a marás számítógéppel segített geometriai szimulációja témakörben, arra a következtetésre jutottam, hogy hardveres gyorsítás alkalmazásával jobb minőségű és lényegesen gyorsabb szimulációs algoritmust lehet kifejleszteni.

Első tézis:

Kidolgoztam egy új heurisztikus marás-szimulációs módszert, ami abban tér el a jelenleg ismert módszerektől, hogy a három- és öttengelyes anyagleválasztás szimulációt, teljes egészében a számítógép videokártyája végzi. Ezzel a módszerrel a szimuláció sebessége és minősége is nagymértékben felgyorsul az eddig ismert eljárásokkal összehasonlítva. A sebességnövekedéssel lehetőség nyílik a munkatér valós idejű animációjára is, amely az öttengelyes megmunkálások szimulációja esetében nélkülözhetetlen.

(disszertáció 2. fejezet)

* * *

Második tézis:

Az első tézisben bemutatott módszer alkalmazásával és a helyességének bemutatásához kidolgoztam egy prototípus marás szimulátor alkalmazást. A szoftver főbb szolgáltatásai:

- Szimulálja egy két körasztalos felépítésű szerszámgép munkatérét.
- Valós időben számítja és kijelzi a megmunkálógép egyes alkatrészeinek egymással való esetleges ütközését az ún. "AABB-tree" módszerrel. Továbbá nullpontfelvételnél is ezt az ütközésvizsgálati eljárást alkalmazza.
- Az anyagleválasztási intenzitás mérése az egyes szerszámelmozdulások között mérhető. Az előre megadott maximális intenzitás átlépése esetén a szimulátor az eltolás értéket automatikusan csökkenti.

- Ötengelyes megmunkálás esetén szerszámhelyzet transzformációt végez a munkadarab koordinátarendszeréből a szerszámgép munkaterének koordinátarendszerébe.
- Geometriai korrekciót (linearizációt) végez. A körasztal és a munkadarab geometriai helyzetéből adódó hibát, a rendszer automatikusan számítja és korrigálja.
- Feldolgozza a tervezett felület geometriáját, és összehasonlítja a 'megmunkált' felülettel.

(disszertáció 6. fejezet)

* * *

A forgácsolási folyamat műveletelemeinek tervezésekor az optimális technológiai paramétereinek meghatározása a gazdaságosság és termelékenység szempontjából kulcsfontosságú feladata. A technológiai paraméterek optimalására alkalmazható matematikai modellnek három összetevője van:

- (1) Az adott feladat mértékadó technológiai korlátainak meghatározása;
- (2) Az optimum-kritériumként használt célfüggvény meghatározása;
- (3) A szerszáméltartam összefüggés, amely a célfüggvény feltételi egyenlete;

A modell független változói, amelyek egyben az optimalizálandó forgácsolási paraméterek, homlokmarás esetén a fogásmélység (d), a fogásszélesség, és az előtolás (f) vagy fogankénti előtolás (f_z).

Harmadik tézis:

Kidolgoztam a homlokmarás anyagleválasztási intenzitáson alapuló technológiai modelljét. Meghatároztam a főbb technológiai korlátokat, korlátfüggvényeket és a költségekvivalens idő függvényét. Megfogalmaztam a CAPP rendszer számára fontos direkt és indirekt feladatot.

(disszertáció 4. fejezet)

* * *

A technológiai gyakorlatban az előírt optimális éltartam és intenzitás a munkadarab geometriájából vagy inhomogenitásából adódóan nem minden esetben érhető el. Szükséges a CAPP rendszerben is alkalmazható modell kialakítása, amely megfelelő módon kezelni tudja ezt a sztochasztikus jelenséget. A kialakított analitikus modell, az optimalizációs módszer alkalmazhatósági határait vizsgálja.

Negyedik tézis:

Analitikus módszerrel megvizsgáltam, milyen hatással van a forgácsoló szerszám éltartamának a technológiai gyakorlatban adódó szórása a relatív költségekvivalens időfüggvény változásra. Két esetet különböztettem meg:

- valamely technológiai korlát az optimalizációs tartományt degenerálja (*degenerált keresési tartomány*), ezért az optimális intenzitás úgynevezett kvázi-optimális lesz.
- nincs mértékadó technológiai korlát (*szabályos keresési tartomány*) ezért az optimális anyagleválasztási intenzitás alkalmazható a műveletelemben

Ha a műveletelemhez tartozó *optimális intenzitás* szabályos keresési tartományban adódott, akkor a relatív éltartam változás, sokkal kedvezőbb befolyással van a relatív költségekvivalens idő változásra, mint ha az *optimális intenzitás* degenerált keresési tartományban adódott volna.

Az analitikus hibavizsgálat fő célja az, hogy segítségével megadhatjuk a megengedett relatív költségeltérés előírásakor a megengedett éltartamszórás tartományát is. Amennyiben ez a szórási tartomány *nagyobb*, mint amely a vizsgált gyártási feladatra a technológiai gyakorlatban valójában előfordul, akkor a *statikus optimalizálás* helyes eredményt ad. Ellenkező esetben viszont, *dinamikus* optimalizáló módszereket kell alkalmazni.

(disszertáció 4. fejezet)

8 A tézisekhez kapcsolódó tudományos publikációk

Szóbeli előadások

- [1] International Workshop on Mechatronics Courses 11-14 June 1997
előadás címe: **Postprocessor development**
- [2] XIII. Szerszámgépek konferencia Miskolci Egyetem 1998. október 26-27,
előadás címe: **Bonyolult felületek előállítása 5 tengelyes posztprocesszor alkalmazásával**
- [3] Doktoranduszok Fóruma, Miskolc, 1998. november 6, Gépészmérnöki Kar szekciója,
előadás címe: **5-tengelyes posztprocesszor fejlesztése**
- [4] microCad '99 International Computer Science Conference February 24-25, 1999,
előadás címe: **Five-axis postprocessor development**
- [5] 2nd International Conference of Phd Students, University of Miskolc, Hungary, 8-14 August 1999, előadás címe: **Advanced Simulation of Five-Axis Machining**
- [6] A magyar tudomány napja, Doktoranduszok Fóruma, Miskolc, 1999. november 4-5. Gépészeti alaptudományok – Gépek és szerkezetek tervezése Szekció
előadás címe: **3-5 tengelyes megmunkálások számítógépes szimulációja**
- [7] microCad '2000 International Computer Science Conference 23-24 February 2000,
előadás címe: **Computer Aided 3-5 Axis Milling Simulation**
- [8] microCad '2000 International Computer Science Conference 23-24 February 2000,
előadás címe: *Cad environment for Machine Tools Design*
- [9] OGÉT 2000, VIII. Országos Gépész Találkozó, 2000. április 7-9. Marosvásárhely
előadás címe: **3 és 5 tengelyes NC megmunkálás számítógépes szimulációja**
- [10] microCad '2001 International Scientific Conference, 1-2 March 2001, előadás címe: **Efficient Simulation of 3-5 Axes Milling**
- [11] microCad '2002 International Computer Scientific Conference 7-8 March 2002, előadás címe: **Cutting Force Modelling in 3-Axis Milling Simulators**

- [12] The Tenth International Conference on Machine Design and Production 4-6 September 2002, Cappadocia, Turkey, előadás címe: *Optimization of Face-Milling Conditions on the Base of Material Removal Rate*
- [13] The Tenth International Conference on Machine Design and Production 4-6 September 2002, Cappadocia, Turkey, előadás címe: *Cutting Force Modeling Possibilities in OpenGL Based Milling Simulators*

Konferencia kiadványokban megjelent cikkek

- [1] Károly Nehéz: *Postprocessor development*, International Workshop on Mechatronics Courses Proceedings 11-14 June p. 208, 1997.
- [2] Nehéz Károly: *Bonyolult felületek előállítása 5 tengelyes posztprocesszor alkalmazásával*, XIII. Szerszámgépek konferencia Miskolci Egyetem B,D,E,F szekciók kiadványa pp. 29-33, 1998. október 26-27.
- [3] Nehéz Károly: *5-tengelyes posztprocesszor fejlesztése*, Doktoranduszok Fóruma, Miskolc, 1998. november 6. Gépészmérnöki Kar szekciókiadványa pp. 48-54.
- [4] Károly Nehéz: *Five-axis postprocessor development*, microCad '99 International Computer Science Conference February 24-25, 1999
Section I: Information Science and Technology, pp. 179-184.
- [5] Károly Nehéz: *Advanced simulation of five-axis machining*, 2nd International Conference of Phd Students, University of Miskolc, Hungary, 8-14 August 1999
pp. 390-395.
- [6] Nehéz Károly: *3-5 tengelyes megmunkálások számítógépes szimulációja*, A magyar tudomány napja, Doktoranduszok Fóruma, Miskolc, 1999. november 4-5. pp. 40-45.
- [7] Károly Nehéz: *Computer Aided 3-5 Axis Milling Simulation*, microCad '2000 International Computer Science Conference 23-24 February 2000, Section I: Information Science and Technology pp.133-138.

- [8] Károly Nehéz, Takács György: *Cad environment for Machine Tools Design*, microCad '2000 International Computer Science Conference 23-24 February 2000, Section I: Information Science and Technology pp.139-144.
- [9] Nehéz Károly, Dr. Csáki Tibor: *3 és 5 tengelyes NC megmunkálás számítógépes szimulációja*, OGÉT 2000, VIII. Országos Gépész Találkozó, 2000. április 7-9. Marosvásárhely, pp. 51-54.
- [10] Károly Nehéz, György Velezdi, Tibor Csáki: *Efficient Simulation of 3-5 Axis Milling*, microCad '2001 Section J: 'Gábor Dénes' Applied Information Technology pp.183-188.
- [11] Károly Nehéz, György Velezdi, Tibor Csáki: *Cutting Force Modelling in 3-Axis Milling Simulators*, microCad '2002 International Computer Scientific Conference 7-8 March 2002. Section: H1 -Applied Information Engineering 2002. pp. 157-162.
- [12] Károly Nehéz, Tibor Tóth: *Optimization of Face-Milling Conditions on the Base of Material Removal Rate*, The Tenth International Conference on Machine Design and Production 4-6 September 2002, Cappadocia, Turkey, pp. 139-152.
- [13] Károly Nehéz, Tibor Csáki: *Cutting Force Modeling Possibilities in OpenGL Based Milling Simulators*, The Tenth International Conference on Machine Design and Production 4-6 September 2002, Cappadocia, Turkey, pp. 127-138.

Folyóiratban megjelent cikkek

- [1] Károly Nehéz, György Velezdi, Tibor Csáki: *Cutting Force Modelling with 3-Axis Milling Simulators*, GÉP, Gépipari tudományos egyesület műszaki folyóirata LIII. évfolyam 2002/2-3 pp. 36-39.
- [2] Károly Nehéz, Tibor Tóth: *Optimization of Face-Milling Conditions on the Base of Material Removal Rate*, Publications of the University of Miskolc, Series C, Mechanical Engineering, journal of Production Systems and Information Engineering, Volume 54. megjelenik decemberben.
- [3] Károly Nehéz, Tibor Csáki: *Cutting Force Modeling Possibilities in OpenGL Based Milling Simulators*, Publications of the University of Miskolc, Series C, Mechanical Engineering, journal of Production Systems and Information Engineering, Volume 54. megjelenik decemberben.

9 Hivatkozások jegyzéke

- [1] Bali János: **Forgácsolás**. Tankönyvkiadó, Budapest 1985, p.150., p.387.
- [2] **Természettudományi Lexikon**. Akadémiai kiadó, Budapest, 1967.
- [3] **Idegen szavak és kifejezések szótára**. Akadémiai kiadó, Budapest, 1989
- [4] Jacobs H.J.: **Optimális forgácsolás**. Műszaki könyvkiadó, Budapest, 1981.
- [5] Gyáni Károly: **A forgácsolás alapjai**. Nemzeti Tankönyvkiadó, 1993., pp. 95-177.
- [6] Káldos E., Nagy E., Takács J.: **Forgácsolás és szerszámai**. Tankönyvkiadó, Budapest 120 p., 1981.
- [7] Horváth Mátyás, Markos Sándor: **Gépgyártástechnológia**. Műegyetemi Kiadó, Budapest 1995.
- [8] Tóth, T., Vadász, D., Perger G.: **A TAUPROG rendszerek jellemzői és az ipari alkalmazás tapasztalatai**. Technos'75. A Gépipari Technológiai Intézet V. tudományos ülésszaka. 1975. pp. 236-249.
- [9] Horváth M., Molnár B. E., Nagy S.: **An attempt in High Level Automation of NC Lates. The FORTAP System, Computer Languages for Numerical Control., Amsterdam-London**. North-Holland Publishing Company, pp 535-547, 1973
- [10] Szegh Imre: **Gyártástervezés**. Műegyetemi kiadó, Budapest, 1996.
- [11] Dudás Illés: **Gépgyártástechnológia I**. Miskolci Egyetemi Kiadó, Miskolc, 2000.
- [12] Bálint Lajos: **A forgácsoló megmunkálás tervezése**. Műszaki könyvkiadó, Budapest, 1961.
- [13] Gribovszki László: **Gépipari megmunkálások**. Tankönyvkiadó Budapest, 1977.
- [14] Tóth Tibor: **Tervezési elvek, modellek és módszerek a számítógéppel integrált gyártásban**. Miskolci Egyetemi Kiadó 1998., p. 208-222.
- [15] Tóth Tibor: **Automatizált műszaki tervezés a gépgyártástechnológiában**. Akadémiai doktori értekezés, Budapest, 1988.
- [16] Sandvik Coromant: **Termelékeny forgácsolás**. 1997.
- [17] Sandvik Coromant: **Modern Metal Cutting**. 1997.
- [18] Milton C. Shaw: **Metal Cutting Principles**. Oxford Science Publications, 1991.
- [19] Detzky, I., Fridik, L., Tóth, T.: **On a New Approach to Computerized Optimization of Cutting Conditions**. Proceedings of the 2nd WorldBasque Congress, Advanced Technology and Manufacturing Conference, Bilbao. Vol. 1., pp. 129-141.

- [20] Bakondi K., Kardos Á.: **A gépgyártás technológiája: I. Forgácsolás.** Tankönyvkiadó, Budapest, 1963.
- [21] Csath Magdolna (szerk.): **Operációkutatás.** Számítástechnikai Oktató Központ,, Budapest, 1972.
- [22] Vadász Dénes: **Tengely- és tárcsaszerű alkatrészek műveleti sorrendtervezésének automatizálása.** Kandidátusi értekezés, Budapest 1982.
- [23] RAYEGANI F.: **Computerized Process Planning and Production Scheduling for Discrete Manufacturing,** Ph.D. Thesis, University of Miskolc, Institute of Information Science, Hungary, 154 p., 1999.
- [24] Dadvandipour Samad: **Solving some optimization problems of computer aided process planning in CIM-environment.** Ph.D. Thesis, University of Miskolc, Institute of Information Science, Hungary, 120 p., 2001.
- [25] Nehéz Károly: **Postprocessor Development.** M.Sc. Thesis, University of Miskolc, Department of Machine Tools, Hungary, 1997.
- [26] Choi, B.K., Jerard R.B.: **Sculptured Surface Machining - Theory and applications.** Kluwer Academic Publishers, ISBN 0412780208, 1998.
- [27] W. H. P. Leslie., szerk.: **NC alkalmazói kézikönyv.** Budapest, Műszaki Könyvkiadó, 1973. p. 451.
- [28] Szegh Imre: **Gyártástervezés.** Műegyetemi kiadó, Budapest, 1996.
- [29] Tóth Tibor: **Automatizált műszaki tervezés a gépgyártástechnológiában 1. rész.** Tankönyvkiadó, Budapest, 1990.
- [30] Tyukodi Zoltán: **NC grafikus szimulátor fejlesztése 3D megjelenítéssel marási folyamatok támogatására,** Diplomamunka, Alkalmazott Informatikai Tanszék, Miskolc, 2002.
- [31] Bali J., Markos S., Mészáros I., Reith J.: **Szerszámkopás modellezése felügyeleti céllal.** Gépgyártástechnológia, 1987. augusztus XXVII. évfolyam.
- [32] PTC: **Pro/Engineer termék leírás.** <http://www.ptc.com/products/proe/index.htm.en>
- [33] Dr. Szirmay-Kalos László.: **Számítógépes grafika.** pp. 192-194., ComputerBooks, Budapest, 1999
- [34] Obádovics J. Gy., Szarka Z.: **Felsőbb matematika** – Scholar kiadó, Budapest p. 381, 1999.
- [35] OpenGL grafikus rendszer hivatalos weboldala: www.opengl.org
- [36] NVIDIA: vezető videó hardvergyártó cég hivatalos weboldala: developer.nvidia.org

- [37] NVIDIA: **Using P-Buffers for Off-Screen Rendering in OpenGL**.
www.developer.nvidia.com, 2001.
- [38] Bergen G.: **Efficient Collision Detection of Complex Deformable Models using AABB Trees**, Journal of Graphics Tools 2 (4), 1997., pp. 1-13.
- [39] ISO 4343:2000 szabvány - **Industrial automation systems -- Numerical control of machines -- NC processor output -- Post processor commands**. p. 271.
- [40] **Standard Template Library Online Reference**. STL generikus osztálykönyvtár leírása,
www.cs.rpi.edu/projects/STL/htdocs/stl.html
- [41] Northwood Designs, Inc: **MetaCut Finish** marás-szimulátor szoftver hivatalos weblapja,
www.metacut.com
- [42] Avgoustinov N.: **Virtual Shaping and Virtual Verification of NC-Programs**, Proceedings of the 31th CIRP International Seminar on Manufacturing Systems, Berkeley, USA, 1998.
- [43] Woo, N., Neider, J., Davis, T., Shreiner, D.: **OpenGL® Programming Guide: The Official Guide to Learning OpenGL, Version 1.2**. Third Edition, Addison Westley, 1999.
- [44] Tóth Tibor, Vadász Dénes: **Az optimális újraélezési szám meghatározása kísérleti-analitikai módszerrel kráteres szerszámkopás esetében**. Műszaki Tudomány 40/3-4. 1969. pp. 208-224.
- [45] Prof. Dr. Ing. W. König, Dipl. Ing. W.R. Depiereux: **Wie lassen sich Vorschub und Schnittgeschwindigkeit optimieren?** Industrie-Anzeiger, Nr. 61.91.Ig.-1- 22. Juli 1969. pp. 17-20.
- [46] Dipl. Ing. K. Essel, Dipl. Ing. E. Götz, et al.: **Adaptive Control -Möglichkeiten zur optimalen Nutzung von Werkzeugmaschinen**. Industrie-Anzeiger, Nr. 60., 20. Juli 1971. pp. 1530-1538.
- [47] Ravignani, G.L., Tipnis, V.A., Friedman, M.Y.: **Cutting Rate – Tool Life Functions (R-R-F)**. General Theory and Applications. Annals of the CIRP Vol. 25/1, 1977., pp. 295-301.
- [48] Somló, J., Nagy, J.: **A New Approach to Cutting Data Optimization. Advances in CAM**. Third IFAC Conference, PROLOMAT, 1976., pp. 293-303.
- [49] Horváth Mátyás, Somló János: **A forgácsolási megmunkálások optimalizálása és adaptív szabályozása**. Műszaki könyvkiadó, Budapest, 1984.
- [50] Kiliç, S.E., Cogun, C., Sen, D.T.: **A Computer-aided Graphical Technique for the Optimization of Machining Conditions**. Computers in Industry, Elsevier (22), 1993., pp. 319-326.
- [51] Tóth, T., Erdélyi, F.: **The inner structure of Computer Aided Process Planning Having Regard to Concurrent Engineering**. Second International Workshop on

Learning in Intelligent Manufacturing Systems, Budapest, Hungary, April 20-21., 1995., pp. 141-161.

- [52] Tóth, T., Erdélyi, F., Rayegani, F.: **CAPP-PPS Integration on the Base of Intensity of Cutting Technology Processes**. Publication of the University of Miskolc Series C. Vol. 45., 1995., pp. 161-169.
- [53] Tóth, T., Erdélyi, F.: **The Role of Optimization and Robustness in Planning and Control of Discrete Manufacturing Processes**. Proceedings of the 2nd World Congress on Intelligent Manufacturing Processes & Systems (edited by László Monostori), Budapest, Hungary, June 10-13., 1997., Springer, pp.205-210.

Katalógusok:

- [54] Sandvik Coromant: **Tools for Die and Mould Makers** – Tool Catalog, 2000., p. 114.
- [55] Sandvik Coromant: **Homlokmarás** – alkalmazási útmutató, C-1120:1 HUN 1998 p. 78.
- [56] Oberg, Jones, Horton, Ryffel, Green, McCauley: **Machinery's Handbook**. ISBN 0-8311-2635-3, 2000.
- [57] Rábel György., szerk.: **Gépipari technológusok kézikönyve**. 3. átdolgozott kiadás, Budapest, Műszaki könyvkiadó, 1984, 978 p.

Forgácsolási erő meghatározása marás esetén:

- [58] Martelloti M.E.: **An analysis of the milling process**. Trans. ASME 63 677-700, 1941.
- [59] Tlusty, J., MacNeil, P.: **Dynamics of cutting forces in end milling**. Annals of the CIRP, 24(1), 1975.
- [60] Kline, W.A., DeVor, R.E., Lindberg, J.R.: **The prediction of cutting forces in end milling with application to concerning cuts**. Int. J. of Machine Tool Design and Research, Vol. 22 Num 1., 1982., pp. 7-22.
- [61] Ersoy, C.: **Automatic Generation of FeedRates for Three Axis End Mill Process Planning**. MSc Thesis, Mechanical Engineering, University of New Hampshire, 1993.
- [62] Spence A.D., Alintas, Y.: **A solid modeller based milling process simulation and planning system**. ASME Journal of Engineering for Industry, 116, Feb., 1994., pp. 61-70.
- [63] P.Lee, Y.Alintas: **Prediction of Ball-end Milling Forces from Orthogonal Cutting Data**. Int. J. Mach. Tools Manufacture Vol. 36. No.9, 1995., pp. 1059-1072.

- [64] His-Yung Feng, Chia-Hsiang Menq: **The Prediction of Cutting Forces in the Ball-End Milling Process-I. Model Formulation and Model Building Procedure.** Int. J. Mach. Tools Manufacture Vol. 34. No.5, 1994., pp. 697-710.
- [65] His-Yung Feng, Chia-Hsiang Menq: **The Prediction of Cutting Forces in the Ball-End Milling Process-II. Cut Geometry Analysis and Modell Verification.** Int. J. Mach. Tools Manufacture Vol. 34. No.5, 1994., pp. 711-719.
- [66] Ching-Chih Tai, Kuang-Hua Fuh: **Model for Cutting Forces Prediction in Ball-End Milling.** Int. J. Mach. Tools Manufacture Vol. 35. No.4, 1994., pp. 511-534.
- [67] H.Z. Li, X.P. Li: **Milling Force Prediction Using a Dynamic Shear Length Model.** Int. J. Mach. Tools Manufacture Vol. 42., 2001., pp. 277-286.
- [68] Engin S., Altintas Y.: **Mechanics and Dynamics of General Milling Cutters. Part I: Helical End Mills.** Int. J. Mach. Tools Manufacture Vol. 41, 2001., pp. 2195-2212.
- [69] Engin S., Altintas Y.: **Mechanics and Dynamics of General Milling Cutters. Part II: Inserted cutters.** Int. J. Mach. Tools Manufacture Vol. 41, 2001., pp. 2213-2231.
- [70] Yan X., Shirase K., Hiaro M., Yasui T.: **Extraction and Optimisation of Cutting Conditions for 2.5-axis milling.** International Journal of Advanced Manufacturing Technology Vol. 16, 2000., pp. 695-699.

Marás-szimuláció:

- [71] Anderson R.O.: **Detecting and eliminating collisions in NC machining.** Computer-Aided Design, Vol. 10, Num. 4, 1978., pp. 231-238.,
- [72] Gossard D.C., Tsuchiya F.S.: **Application of set theory to the verification of NC tapes.** Proc. North American Metalworking Conference, April, 1978.
- [73] Voelcker H.B., Hunt W.A.: **The role of solid modeling in machining.** Process Modeling and NC Verification, SAE Technical Paper 810195, 1981.
- [74] Chappel Y.T.: **The use of vectors to simulate material removed by numerically controlled milling.** Computer-Aided Design, Vol. 15, Num. 3, 1983.
- [75] Wang W.P., Wang K.K.: **Real-time verification of multiaxis NC programs with raster graphics.** Proceedings of the IEEE Intl. Conf. on Robotics and Automation, San Francisco, April 7-10., 1986., pp.166-171.
- [76] Hook van Y.: **Real time shaded NC milling display.** Computer Graphics, Vol. 20, No 4. (Proc. SIGGRAPH'86), pp.15-20.
- [77] Atherton P.R., Earl C. Fred C.: **A graphical simulation system for dinamic five-axis NC verification.** Proc. Autofact (Nov.), SME, Dearborn, Mi., 1987., 2.1-2.12.

- [78] Jerard R.B., Hussaini S.Z., Drysdale R.L. Schaudt B.: **Approximate methods for simulation and verification of numerically controlled machining programs.** The Visual Computer, Vol. 5, Num. 6, 1989.
- [79] Drysdale R.L., Jerard R.B., Schaudt B., Hauck K.: **Discrete simulation of NC Machining.** Algorithmica Vol. 4, 1989., pp. 33-60.
- [80] Jerard R.B., Angleton J.M. Drysdale R.L., Su P.: **The use of surface points sets for generation, simulation, verification and automatic correction of NC machining programs.** Proc. NSF Des. and Manuf. Sys. Conf., SME USA, 1989., pp. 143-151.
- [81] Oliver J.H. Goodman E.D.: **Direct dimensional NC verification.** Computer-Aided Design, 22 (1), 1990., pp. 3-10.
- [82] Saito T., Takahashi T.: **NC machining with G-Buffer method.** Computer Graphics, Vol. 25, No. 4, Proc. SIGGRAPH'91, 1991., pp. 207-216.
- [83] Oliver J.H.: **Efficient intersection of surface normals with milling tool swept volumes for discrete three-axis NC verification.** ASME Journal of Mechanical Design, June 14., 1992., pp. 283-281.
- [84] Kawashima, Y., Itoh K., Ishida T.: **A flexible quantitative method for NC machining verification using a space-division based solid model.** The Visual Computer 7, 1991., pp. 149-157.
- [85] Lindstorm P., Koller D. Ribarsky W., Hodges L. F.: **Real-Time, Continuous Level of Detail Rendering of Height Fields.** Proceedings of ACM SIGGRAPH 96, August, 1996., pp. 109-118.
- [86] Yang M. Lee E.: **NC verification for wire-EDM using an R-map.** Computer Aided Design, Vol. 28, No. 9, 1996., pp. 733-740.
- [87] Souring A.I., Pasko A.A.: **Function representation for sweeping by a moving solid.** IEEE Transactions on Visualisation and Computer Graphics, Vol. 2, No. 2, 1996., pp. 11-18.
- [88] Dudás László: **Kapcsolódó felületepárok gyártásgeometriai feladatainak megoldása az elérés modell alapján.** Kandidátusi értekezés, Budapest, TMB, 1991. p.144.
- [89] László Dudás: **A Consistent Model for Generating Conjugate Surfaces and Determining All the Types of Local Undercuts and Global Cut.** UMTIK'96 International Machine Design and Production Conference, Ankara 1996. sept. 11-13. pp.467-476.
- [90] Roy U., Xu Y.: **Computation of a geometric model of a machined part from its NC machining programs.** Computer-Aided Design 31, 1999., pp. 401-411.

- [91] Albersmann F., Zabel A., Müller H., Weller F.: **Efficient Direct Rendering of Digital Height Fields.** CSI International Conference on Visual Computing (ICVC'99) Goa, India, Febr. 23-26., 1999., pp. 44-52.
- [92] Müller H., Friedhoff J., Weinert K.: **Efficient Discrete Simulation of 3-axis Milling.** Technical report No. 591, University of Dortmund, 1995.
- [93] Inui M., Kaneda M., Kakio R.: **Fast Simulation of Sculptured Surface Milling with 3-Axis NC Machine.** IPSJ JOURNAL Contents, Vol.40, No.04 – 051, 1998.
- [94] König A., Gröller E.: **Real Time Simulation and Visualization of NC Milling Processes for Inhomogeneous Materials on Low-End Graphics Hardware.** Technical Report TR-186-2-98-04, Institute of Computer Graphics, Vienna University of Technology, January, 1998
- [95] Glaeser G., Gröller E.: **Efficient Volume-Generation During the Simulation of NC-Milling.** Mathematical Visualisation, Springer Verlag, Heidelberg, 1998., pp. 89-106.
- [96] Chiou J., Lee S.: **A shape-generating approach for multi-axis machining G-buffer models.** Computer-Aided Design 31, 1999., pp. 761-776.
- [97] Yunching Huang, Oliver J.H.: **NC Milling Error Assessment and Tool Path Correction.** Proceedings of the 21st annual Conference on Computer Graphics, 1997., pp. 287-294.
- [98] Müller H., Albersmann F., Zabel A.: **Efficient Raster-Based Simulation and Visualization of 3-Axis Milling of Free-Formed Shapes.** Research Report N.o. 667/1998, Fachbereich Informatik, Universität Dortmund, Germany.
- [99] Hsu P-L., Yang W-T.: **Realtime 3D simulation of 3-axis milling using isometric projection.** Computer-Aided Design, Volume 25, Number 4, 1993., pp.215-224,
- [100] Liping Wang, Ming C. Leu, Denis Blackmore: **Generating Swept Solids for NC Verification the SEDE Method.** SMA '97, Proceedings of the Fourth Symposium on Solid Modeling and Applications, May 14-16., 1997., Atlanta, Georgia, USA, pp. 364-375.
- [101] Frey D.D., Otto K. N., Pflanger W.: **Swept Envelopes of Cutting Tools in Integrated Machine and Workpiece Error Budgeting.** Annals of the CIRP, Vol. 46, No. 1, 1997., pp. 475-480.
- [102] Chung Yun C., Park Jung. W., Shin Hayon, Choi Byoung K.: **Modeling the surface swept by a generalized cutter for NC verification.** Computer-Aided Design Volume. 30 number 8, 1998., pp. 587-594.
- [103] Blackmore D., Leu M. C., Wang L. P.: **The sweep-envelope differential equation algorithm and its application to NC machining verification.** Computer-Aided Design, Vol. 29, Num. 9, 1997., pp. 629-637.

- [104] Jang D., Kim K., Jung J.: **Voxel-Based Virtual Multi-Axis Machining**. International Journal of Manufacturing Technology Vol. 16, 2000., pp 709-713.
- [105] Theoharis T., Papaioannou G., Karabassi E.: **The Magic of the Z-buffer: a survey**. Department of Informatics, University of Athens, WSCG Conference Proceedings, 2001.
- [106] Stewart N., Leach G., John S.: **An Improved Z-Buffer CSG rendering Algorithm**. <http://citeseer.nj.nec.com/478488.html>, 2000.
- [107] CGTech:VeriCUT dokumentáció: www.vericut.com
- [108] MetaCut szoftver hivatalos honlapja www.nwdesigns.com/home/Finish/Overview.htm
- [109] Srivastava A. K., Veldhus S. C., Elbestawit M. A.: **Modelling geometric and thermal errors in a five-axis CNC machine tool**. International Journal of Machine Tools and Manufacturing Volume 35. No. 9., 1995., pp. 1321-1337.
- [110] Mahbubur R. MD., Heikkala J., Lappalainen K., Karjalainen J. A.: **Positioning accuracy improvement in five-axis milling by post processing**. International Journal of Machine Tools and Manufacturing Volume 37. No. 2., 1997., pp. 223-236.
- [111] Takeuchi Y., Watanabe T.: **Generation of 5-axis control collision-free tool path and post processing for NC data**. Annals of the CIRP Vol. 41/1, 1992., pp. 539-542.

10 Ábrajegyzék

1. ábra: az NC megmunkáláskor keletkező hibák csoportosítása.	16
2. ábra: a marószerszámok származtató felületeinek kétváltozós függvényei.	18
3. ábra: a diszkrét vektor módszer két alaptípusa.	20
4. ábra: a legszélesebb körben alkalmazott z-map modell elvi alapja.	21
5. ábra: Atherton által javasolt z-map megvalósítás.	22
6. ábra: Van Hook kiterjesztett z-map módszere.	23
7. ábra: a nyolc bemutatott megmunkálás-szimulációs eljárás.	26
8. ábra: a diszkrét módszerek pontossági vizsgálatához szükséges modellek.	27
9. ábra: a mintavételezési pontok távolsága.	28
10. ábra: a „ Γ -buffer” eljárás részbeni megvalósításával készült szimulátor.	29
11. ábra: az OpenGL egyszerűsített architektúrája.	30
12. ábra: a z-buffer eljárás működése.	32
13. ábra: az új p-buffer eljárás magja.	33
14. ábra: marás-szimuláció a p-bufferben.	35
15. ábra: legegyszerűbb buffer-megjelenítési algoritmus.	36
16. ábra: két körasztallal megvalósított 5-tengelyes gépkonfiguráció modellje.	37
17. ábra: a linearizáció folyamat elvi modellje.	40
18. ábra: a hierarchikus folyamattervezési modell, globális és lokális tervezési szintjei.	44
19. ábra: homlokmarás a jellemző technológiai jellemzőivel (oldal és felülnézetben).	47
20. ábra: optimalizálási tartomány a (Q, R, τ) állapotterben, a mértékadó korlátok feltüntetésével.	54
21. ábra: optimalizálási tartomány a (Q, R, τ) állapotterben (kinagyítva).	55
22. ábra: degenerált keresési tartomány magas éltartamkorlát esetén.	56
23. ábra: degenerált keresési tartomány alacsony teljesítményű szerszámgép miatt.	57
24. ábra: az indirekt feladat néhány tipikus döntési esete.	58
25. ábra: a fajlagos költségekvalens idő függvény ábrázolása.	62
26. ábra: a költségekvalens időfüggvény, mint egyváltozós függvény jellegh. ábrázolása.	64
27. ábra: a relatív költségekvalens idő ábrázolása ábrázolása egy konkrét példa esetén.	67
28. ábra: a relatív költségekvalens idő ábrázolása szabályos keresési tartomány esetén.	69
29. ábra: az éltartamfüggvény meghatározása többváltozós lineáris regresszióval.	71
30. ábra: ellenirányú és egyirányú maráskor keletkező F_c erő iránya és nagysága.	75
31. ábra: szerszámmozgás A pontból B pontba a "z-map" felett.	78
32. ábra: az térfogatmodell hiányosságát szemléltető ábra.	79
33. ábra: pillanatnyi forgácsvastagság.	80
34. ábra: a valóságos forgácsvastagság a felületi normális és az előtolás vektorok függvénye.	80
35. ábra: az érintkezési tartomány értelmezése.	81
36. ábra: a gömbvégű marószerszám modellje élgeometria szempontjából.	82

37. ábra: modellben használt szögek rendszere.	83
38. ábra: koordináta transzformáció a szerszám általános elmozdulása esetén.	84
39. ábra: szerszámelhajlásból adódó felületi hiba 5 különböző szerszámhelyzetben.	87
40. ábra: a fellépő erők analitikus modellezése a p-bufferes szimulátorban.	88
41. ábra: a szimulátor szoftver főbb megvalósított osztályai.	90
42. ábra: a szimulátor funkcionális modellje.	91
43. ábra: a szerszámpálya betöltése után megjelenő dialógusablak.	92
44. ábra: a geometriai szimuláció egy köztes fázisa.	93
45. ábra: az anyagleválasztás intenzitásának mérése.	94
46. ábra: a kész felület pontosságának vizsgálata félig átlátszó síkfelület mentén.	95
47. ábra: az öttengelyes példában felhasznált szabadformájú felületet ábrázolja.	96
48. ábra: az öttengelyes munkatérbeli anyagleválasztás szimuláció egy köztes fázisa.	97
49. ábra: a kész munkadarab a teljes szerszámpálya szimuláció után.	98
50. ábra: a p-bufferes eljárás sebességének összehasonlító eredményei.	99

Első melléklet: Több-változós nemlineáris regresszió

A (78)-ban szereplő egyenletet logaritmus transzformációval a következő alakra hozzuk:

$$H^* = \sum_{i=1}^n \left(\lg T_i^m - \lg \frac{C_v D_{c_i}^{w_v}}{d_i^{x_v} f_{z_i}^{y_v} w_i^{z_v} z_i^{p_v} v_i} \right)^2 \Rightarrow \min. \quad (106)$$

Meg kell határozni, n darab mérési adat esetén, az ismeretlen kitevőket azzal a feltétellel, hogy H^* minimum legyen. Ehhez a következő számítási lépések szükségesek: először előállítjuk H^* minden ismeretlen változója szerinti parciális deriváltjait, majd egyenként zérussal egyenlővé tesszük őket, és megoldjuk a kiadódó hét ismeretlenes lineáris egyenletrendszer a keresett változókra.

$$\frac{\partial H^*}{\partial C_v} = \frac{-2}{C_v} \sum_{i=1}^n \left(\lg T_i^m - \lg \frac{C_v D_{c_i}^{w_v}}{d_i^{x_v} f_{z_i}^{y_v} w_i^{z_v} z_i^{p_v} v_i} \right) = \quad (107)$$

$$m \sum_{i=1}^n \lg T_i - n \lg C_v - w_v \sum_{i=1}^n \lg D_{c_i} + x_v \sum_{i=1}^n \lg d_i + y_v \sum_{i=1}^n \lg f_{z_i} + z_v \sum_{i=1}^n \lg w_i + p_v \sum_{i=1}^n \lg z_i + \sum_{i=1}^n \lg v_i = 0.$$

$$\begin{aligned} \frac{\partial H^*}{\partial x_v} &= 2 \sum_{i=1}^n \left(\lg d_i \left(\lg T_i^m - \lg \frac{C_v D_{c_i}^{w_v}}{d_i^{x_v} f_{z_i}^{y_v} w_i^{z_v} z_i^{p_v} v_i} \right) \right) = m \sum_{i=1}^n \lg T_i \lg d_i - \lg C_v \sum_{i=1}^n \lg d_i - w_v \sum_{i=1}^n \lg D_{c_i} \lg d_i \\ &+ x_v \sum_{i=1}^n (\lg d_i)^2 + y_v \sum_{i=1}^n \lg f_{z_i} \lg d_i + z_v \sum_{i=1}^n \lg w_i \lg d_i + p_v \sum_{i=1}^n \lg z_i \lg d_i + \sum_{i=1}^n \lg v_i \lg d_i = 0. \end{aligned} \quad (108)$$

$$\begin{aligned} \frac{\partial H^*}{\partial y_v} &= m \sum_{i=1}^n \lg T_i \lg f_{z_i} - \lg C_v \sum_{i=1}^n \lg f_{z_i} - w_v \sum_{i=1}^n \lg D_{c_i} \lg f_{z_i} + x_v \sum_{i=1}^n \lg d_i \lg f_{z_i} + y_v \sum_{i=1}^n (\lg f_{z_i})^2 \\ &+ z_v \sum_{i=1}^n \lg w_i \lg f_{z_i} + p_v \sum_{i=1}^n \lg z_i \lg f_{z_i} + \sum_{i=1}^n \lg v_i \lg f_{z_i} = 0. \end{aligned} \quad (109)$$

$$\begin{aligned} \frac{\partial H^*}{\partial z_v} &= m \sum_{i=1}^n \lg T_i \lg w_i - \lg C_v \sum_{i=1}^n \lg w_i - w_v \sum_{i=1}^n \lg D_{c_i} \lg w_i + x_v \sum_{i=1}^n \lg d_i \lg w_i + y_v \sum_{i=1}^n \lg f_{z_i} \lg w_i + \\ &z_v \sum_{i=1}^n (\lg w_i)^2 + p_v \sum_{i=1}^n \lg z_i \lg w_i + \sum_{i=1}^n \lg v_i \lg w_i = 0. \end{aligned} \quad (110)$$

$$\begin{aligned} \frac{\partial H^*}{p_v} &= m \sum_{i=1}^n \lg T_i \lg z_i - \lg C_v \sum_{i=1}^n \lg z_i - w_v \sum_{i=1}^n \lg D_{c_i} \lg z_i + x_v \sum_{i=1}^n \lg d_i \lg z_i + y_v \sum_{i=1}^n \lg f_{z_i} \lg z_i + \\ & z_v \sum_{i=1}^n \lg w_i \lg z_i + p_v \sum_{i=1}^n (\lg z_i)^2 + \sum_{i=1}^n \lg v_i \lg z_i = 0. \end{aligned} \quad (111)$$

$$\begin{aligned} \frac{\partial H^*}{w_v} &= m \sum_{i=1}^n \lg T_i \lg D_{c_i} - \lg C_v \sum_{i=1}^n \lg D_{c_i} - w_v \sum_{i=1}^n (\lg D_{c_i})^2 + x_v \sum_{i=1}^n \lg d_i \lg D_{c_i} + y_v \sum_{i=1}^n \lg f_{z_i} \lg D_{c_i} + \\ & z_v \sum_{i=1}^n \lg w_i \lg D_{c_i} + p_v \sum_{i=1}^n \lg z_i \lg D_{c_i} + \sum_{i=1}^n \lg v_i \lg D_{c_i} = 0. \end{aligned} \quad (112)$$

$$\begin{aligned} \frac{\partial H^*}{m} &= m \sum_{i=1}^n (\lg T_i)^2 - \lg C_v \sum_{i=1}^n \lg T_i - w_v \sum_{i=1}^n \lg D_{c_i} \lg T_i + x_v \sum_{i=1}^n \lg d_i \lg T_i + y_v \sum_{i=1}^n \lg f_{z_i} \lg T_i + \\ & + z_v \sum_{i=1}^n \lg w_i \lg T_i + p_v \sum_{i=1}^n \lg z_i \lg T_i + \sum_{i=1}^n \lg v_i \lg T_i = 0. \end{aligned} \quad (113)$$

A hét egyenlet felírható mátrix alakban is ($\mathbf{Ax} = \mathbf{b}$):

$$\begin{bmatrix} \sum_{i=1}^n \lg T_i & -\sum_{i=1}^n \lg D_{c_i} & \sum_{i=1}^n \lg d_i & \sum_{i=1}^n \lg f_{z_i} & \sum_{i=1}^n \lg w_i & \sum_{i=1}^n \lg z_i & -n \\ \sum_{i=1}^n \lg T_i \lg d_i & -\sum_{i=1}^n \lg D_{c_i} \lg d_i & \sum_{i=1}^n (\lg d_i)^2 & \sum_{i=1}^n \lg f_{z_i} \lg d_i & \sum_{i=1}^n \lg w_i \lg d_i & \sum_{i=1}^n \lg z_i \lg d_i & -\sum_{i=1}^n \lg d_i \\ \sum_{i=1}^n \lg T_i \lg f_{z_i} & -\sum_{i=1}^n \lg D_{c_i} \lg f_{z_i} & \sum_{i=1}^n \lg d_i \lg f_{z_i} & \sum_{i=1}^n (\lg f_{z_i})^2 & \sum_{i=1}^n \lg w_i \lg f_{z_i} & \sum_{i=1}^n \lg z_i \lg f_{z_i} & -\sum_{i=1}^n \lg f_{z_i} \\ \sum_{i=1}^n \lg T_i \lg w_i & -\sum_{i=1}^n \lg D_{c_i} \lg w_i & \sum_{i=1}^n \lg d_i \lg w_i & \sum_{i=1}^n \lg f_{z_i} \lg w_i & \sum_{i=1}^n (\lg w_i)^2 & \sum_{i=1}^n \lg z_i \lg w_i & -\sum_{i=1}^n \lg w_i \\ \sum_{i=1}^n \lg T_i \lg z_i & -\sum_{i=1}^n \lg D_{c_i} \lg z_i & \sum_{i=1}^n \lg d_i \lg z_i & \sum_{i=1}^n \lg f_{z_i} \lg z_i & \sum_{i=1}^n \lg w_i \lg z_i & \sum_{i=1}^n (\lg z_i)^2 & -\sum_{i=1}^n \lg z_i \\ \sum_{i=1}^n \lg T_i \lg D_{c_i} & -\sum_{i=1}^n (\lg D_{c_i})^2 & \sum_{i=1}^n \lg d_i \lg D_{c_i} & \sum_{i=1}^n \lg f_{z_i} \lg D_{c_i} & \sum_{i=1}^n \lg w_i \lg D_{c_i} & \sum_{i=1}^n \lg z_i \lg D_{c_i} & -\sum_{i=1}^n \lg D_{c_i} \\ \sum_{i=1}^n (\lg T_i)^2 & -\sum_{i=1}^n \lg D_{c_i} \lg T_i & \sum_{i=1}^n \lg d_i \lg T_i & \sum_{i=1}^n \lg f_{z_i} \lg T_i & \sum_{i=1}^n \lg w_i \lg T_i & \sum_{i=1}^n \lg z_i \lg T_i & -\sum_{i=1}^n \lg T_i \end{bmatrix} \begin{bmatrix} m_v \\ w_v \\ x_v \\ y_v \\ z_v \\ p_v \\ \lg C_v \end{bmatrix} = \begin{bmatrix} -\sum_{i=1}^n \lg v_i \\ -\sum_{i=1}^n \lg v_i \lg d_i \\ -\sum_{i=1}^n \lg v_i \lg f_{z_i} \\ -\sum_{i=1}^n \lg v_i \lg w_i \\ -\sum_{i=1}^n \lg v_i \lg z_i \\ -\sum_{i=1}^n \lg v_i \lg D_{c_i} \\ -\sum_{i=1}^n \lg v_i \lg T_i \end{bmatrix} \quad (114)$$

Az egyenlet valamilyen numerikus módszer segítségével nagy pontossággal megoldható.

Második melléklet: Minta-adatbázis (elvi modell)

[illegible]

