

STEP fájl alapú automatikus gyártási időbecslés

Szerzők:

Berencsi Dániel (LKCNP1)

Pördi Patrik Dominik (CSJJVA),

Tanicsa Viktor (EDKZI0)

Konzulens:

Kutrovács Lajos

Mesteroktató

Budapest, 2022

Tartalom

I	Bevezetés.....	1
I.1	Alaphelyzet.....	1
I.2	Feladatunk és célkitűzésünk	2
II	A STEP fájlok alapja.....	6
II.1	Mi is a STEP	6
II.2	A STEP fájlok felépítése	7
II.3	A STEP fájlok feldolgozása	11
III	Alaksajátosságok.....	13
III.1	Mi is az a feature	13
III.2	Irodalmi áttekintés az alaksajátosságok felismerésének témakörében	16
IV	Az elkészült program	19
IV.1	STEP fájl feldolgozás	19
IV.1.1	Felhasznált elméleti alapok	19
IV.1.2	Megvalósítás.....	20
IV.2	Alaksajátosság felismerés	22
IV.2.1	Felhasznált elméleti alapok	22
IV.2.2	Megvalósítás.....	26
IV.2.3	Program bemutatása	31
V	Gyártási időbecslés alapjai	33
V.1	Irodalomkutatás a feature alapú gyártási időbecslésről.....	33
V.1.1	Becslés megadásának lehetőségei	33
V.1.2	Az általunk használt megfontolás	35
V.2	Szükséges adatok rendszerbe foglalása	36
V.2.1	Út a becsléshez.....	36
V.2.2	A szükséges információk	37
V.3	Adatbázis bemutatása	40
VI	Gyártási időbecslés és felhasználása a gyakorlatban	43
VI.1	Időbecslés különböző megmunkálásokra.....	43
VI.2	Normaidő	47
VI.3	Költségbecslés.....	49
VI.3.1	Költségbecslés a gyakorlatban	49
VII	Mintapélda.....	50
VIII	Konklúzió és jövőbeli fejlesztési lehetőségek.....	55
IX	Irodalomjegyzék.....	57

I Bevezetés

I.1 Alaphelyzet

Manapság a legnagyobb ipari kihívások közé tartozik a hatékonyság maximalizálása, amelynek része az egyes tevékenységekre fordított idő és a felesleges költségek minimalizálása, valamint a folyamatok gyorsítása, megbízhatóságának növelése, azaz automatizálása. A feladat több tényezőtől álló problémakör, hiszen sok szakember, szakterület és környezeti körülmény befolyásolja a megoldásokat. Az életút, amelyet a termék az igény felmerülésétől a gyártás elkezdéséig bejár, egy összetett problémakörön végigvezetett, hosszú és sok erőforrást igénylő iterációs folyamat. A terméktervező elkészít egy konstrukciós modellt, majd a gyártástervező mérnök ellenőrzi a gyárthatóságot, megtervezi a termék előállítás technológiai folyamatát. Már ezen két terület között is felmerülnek folyamatos egyeztetést igénylő körülmények, vitás szituációk és visszatérő problémák. Sokszor előfordul, hogy az adott termék nem gyártható vagy a meglévő infrastruktúra mellett az előállítás túl költséges, nem hatékony vagy nem kivitelezhető. Ezek eredményeképpen a termék többszöri újra tervezésen eshet át, vagy adott esetben az eszközpark bővítése vagy átalakítása is szóba kerülhet. A gyártási feladatok és a termék előállításához szükséges kapacitások összerendelése, elégségességének megteremtése újabb szakterületekkel kapcsolatos egyeztetések igényét veti fel. Már a szakmai kérdések megoldása sem egy egyszerűen kezelhető kérdés, de nem feledkezhetünk meg a management oldalról, akik a változtatások költségvonzatának ismeretében a döntéseket felülbírállhatják. Az ő céljuk, hogy maximalizálják a várható profitot, amely a bevételek növelését és a felhasznált erőforrások, vagyis a költségek minimalizálását jelenti, anélkül, hogy sérülne a végtermék minősége. A szereplők végcélja alapvetően közös, megtalálni az igényeket leghatékonyabban kielégítő, legolcsóbb alternatívát. A cél eléréséhez minden oldalról kompromisszumok szükségesek, melyek kialakulásában a különböző résztvevők szakképzettségéből adódó kommunikációs nehézségek áthidalása elengedhetetlen. A kommunikáció gyorsítását, pontosságának, félreérthetlenségének növelését szolgálják azok az automatizált megoldások, amelyek segítségével az információk emberi beavatkozás nélkül, vagy annak minimalizálásával állíthatók elő.

Az ipar 4.0 azoknak a magasan fejlett, teljesen automatizált gyártórendszereknek az implementálását jelenti, melyek képesek monitorozni és szimulálni a termék teljes életciklusát, az ötlet megszületésétől a tervezésen, gyártás előkészítésen, termelésen, felhasználáson át egészen a megsemmisítésig. Itt a fókusz a folyamatok, berendezések hatékonyságán, az

eszközök által szolgáltatott adatok mérésén, feldolgozásán és az automatikus beavatkozáson van. Azonban az előző bekezdésben tárgyalt emberi tényezőkből adódó bizonytalanság megszüntetése egy olyan nem megoldott témakör, amellyel érdemes foglalkozni. A mi célunk az, hogy egy egyszerű megoldást kínáljunk a fent említett problémakör egy szeletére, kiküszöböljük, illetve csökkentjük az emberi tényező szerepét, valamint hidat teremtünk a szakterületek között. Továbbá a jövőbe tekintve az ipar 5.0 szellemiségét is követnénk, újra definiálva így az ember szerepét a folyamatban. Az itt tárgyaltakhoz hasonló szituációban, az automatizálás teljes kiaknázásával emberi erőforrásokat egy olyan kreatív munkával kapcsolatos irányba szeretnénk terelni, amely hosszútávon egy fenntarthatóbb és ergonomikusabb folyamatot és a vevő igényeit teljesen kielégítő szolgáltatást eredményez.

A gyártási folyamat tervezésében a sok, műszakilag közel azonos értékű megoldás közötti döntés rendszerint időigény, azaz gyorsaság és költség alapon születhet meg. A költség általában a felhasznált erőforrások igénybevételi idejével arányosan jelentkezik, tehát kijelenthető, hogy a termék előállítás megtervezésének különböző területein nagy jelentőségű az adott szinthez szükséges idő adatok gyors és olcsó meghatározása. A módszer legitimitását alkalmazási sokszínűségével lehet validálni. A gyártó vállalatok többsége küzd azzal a problémával, hogy nagyon sok ajánlatot kell elkészíteni, de azoknak csak csekély hányadából lesz megrendelés. Ez a munka a mérnöki erőforrásokat leköti, azaz költséget okoz, bevétel viszont csak a megnyert munkákból lesz. Ha sikerülne kisebb energia befektetéssel gyártási információhoz jutni, ezáltal lehetővé tenni az idő- és költségbecslést, az nagyban megnövelné a mérnöki munka hatékonyságát, és számos egyéb területen is segítheti a munkát. Egy ilyen további alkalmazási lehetőség például, hogy a megrendelő árajánlat kérés előtt a termékről egy becsült előállítási költséget és időt kap, mely viszonyítási alapként szolgálhat a beérkező ajánlatok kontrollálásához.

I.2 Feladatunk és célkitűzésünk

A célunk az volt, hogy képesek legyünk STEP fájlok részleteiben történő megismerésére úgy, hogy megbecsülhessük gyártási értékeit. Az általunk kínált megoldás egy olyan szoftver, amely a munkadarab CAD modellje és egy - korábbi megmunkálási információkat tartalmazó adatbázis segítségével definiált - tapasztalati modell alapján végez gyártási időbecslést, majd belőle várható gyártási költséget állítunk elő.

Az ötletünk lényegében az volt, hogy egy STEP fájl feldolgozása során megtaláljuk a hozzá tartozó összes gyártási alaksajátosságot, azaz feature-t és az őket felépítő felületeket, valamint

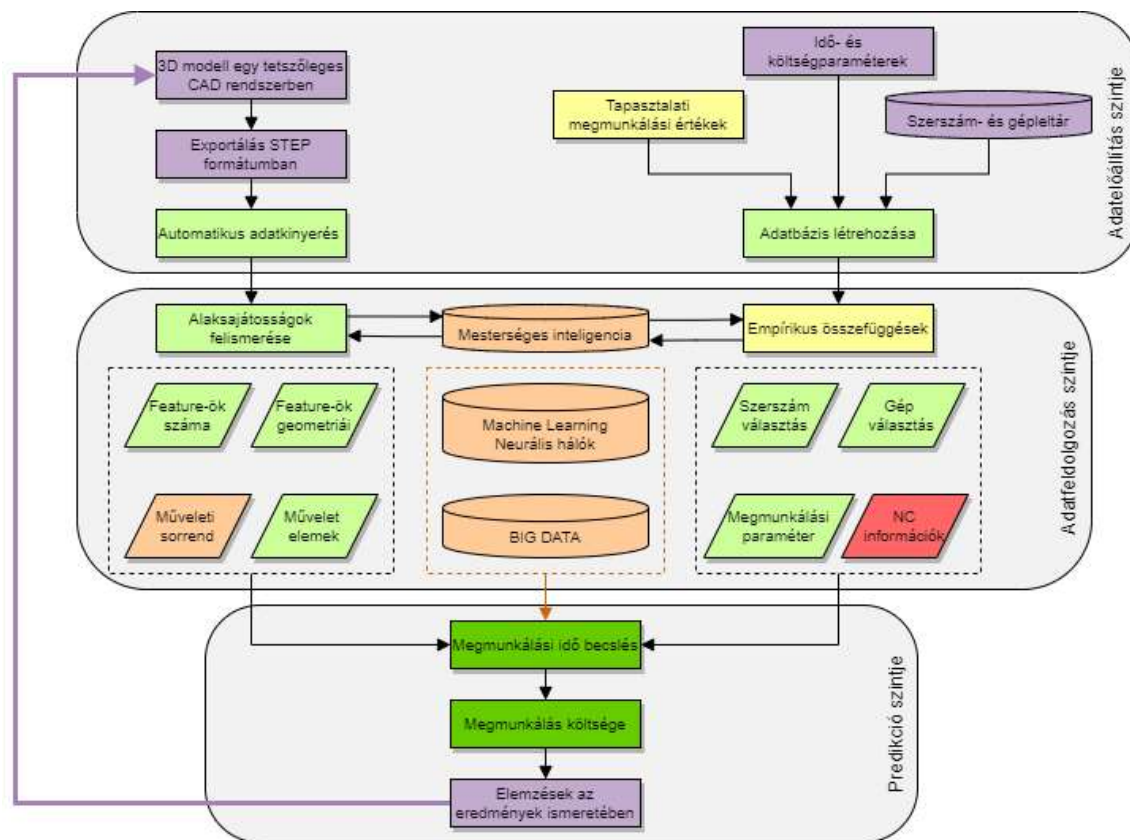
azok geometriai paramétereit. Az adott alaksajátosságokat megmunkálási technológiához tudjuk rendelni, jelen esetben, esztergálásról, marásról és fúrásról beszélhetünk. A megmunkálási technológia és a lemunkálandó anyagmennyiség geometriájának és elhelyezkedésének ismeretében különböző következtetéseket tudunk levonni. Esetünkben a gyártási időbecslés az terület, amire koncentrálni szeretnénk. Tehát, a kinyert információkat összekapcsoljuk empirikus összefüggésekkel, tapasztalati értékekkel, továbbá az eljárásokhoz és technológiához szorosan kapcsolódó különböző átlagos értékekkel, amelyek egy előre definiált adatbázisban vannak tárolva. Végeredményként rendelkezésünkre áll minden olyan szükséges információ, mellyel becsülhető a termék előállításához szükséges idő.

A folyamatot a 1. Ábra szemlélteti, megértést segítő színekkel ellátva, három szintes struktúrában. A színek alapvetően az adott részegységek elkülönülését szimbolizálják. A lila szín a felhasználókhoz köthető be- és kimeneti elemeket jelöli, azaz ezek felhasználási területtől függően tetszés szerint módosíthatóak. Pasztell sárgával a rajtuk felül álló axiómák és összefüggések találhatók. A zöld, a narancssárga és a piros színek, pedig sorban, az általunk létrehozott, a jövőben létrehozandó, illetve nem létrehozandó egységeket jelölik.

Ahogy látható az általunk felépített struktúra három szinttel rendelkezik, ezek pedig, az Adatellőállítás, az Adatfeldolgozás és a Predikció, valamint a dolgozathoz hasonlóan két fő szálon fut. A bal oldali szakasz, az az, a STEP fájl feldolgozása során történő információkinyerés a dolgozat első részében kerül tárgyalásra. Ezen a területen elég mély betekintésre kerül sor az elméleti és programozástechnikai megfontolásokra melyeket használunk. Itt alapvetően az első két szint kerül tárgyalásra, a STEP fájl beolvasása során történő adatkinyerés, valamint a nyers adatok feldolgozásával előállított megmunkálási jellemzők. Ahogy az ábra is szemlélteti, a műveletelemekre és a megmunkálási paraméterekre, az alaksajátosságok geometriájából és számából következtetünk, valamint, hogy jelenleg még nem vagyunk képesek a műveleti sorrend megállapítására.

A jobb oldali szakasz kevésbé elméleti és programozástechnikai területekkel foglalkozik, itt sokkal inkább hangsúlyos a gyártást jellemző értékek és összefüggések segítségével történő becslés előkészítése. Az első szinten az adott körülményeket jellemző adatbázis létrehozása a mérvadó, míg a második szint a benne található információk adott területhez köthető felhasználásával foglalkozik. Ahogy az ábrából is látható, NC információkkal nem szeretnénk foglalkozni, a dolgozat céljai egy tőlük teljesen független becslés előállítása. A struktúra harmadik szintje is ide kapcsolódik, az az a jobb és bal oldal jelenlegi szintézise. A Predikció az alaksajátosságok felismerésén alapuló információkat helyezi gyártási környezetbe, ezáltal

várható idő és költségbecslést létrehozva. Amint a színkódból is látható, a becsült információk ismeretében felhasználói szintű elemzések végezhetőek el, így biztosítva a rendszer be és kimenete közötti zárt láncú összeköttetést, amely végső soron a mindenki igényeit kielégítő termék előállítását teszi lehetővé.



1. Ábra: A felépített struktúra

A két párhuzamosan futó szál az Adatfeldolgozás szintjén található Mesterséges intelligencia segítségével történő összeköttetése még nem került szóba. A dolgozat teljessége érdekében, valamint kiméretének figyelembevételével a különböző gépi tanuláshoz köthető megoldásokkal nem foglalkoztunk, viszont a jövőben prioritásként gondolunk rá. Következésképpen, a program Python környezetben íródott, amely egy interpretált magasabb rendű objektum orientált programozási nyelv, mely egyre nagyobb teret hódít. Azért döntöttünk ezen nyelv használata mellett mert rengeteg olyan előnyt tud felmutatni, melyet konkurensei nem. Egyrésről viszonylag egyszerű és egyszerűen olvasható szintaktikával rendelkezik, másrésről olyan számolást és szövegfeldolgozást segítő könyvtárakat tartalmaz, melyek nagyban megkönnyítették a programírás korai szakaszát. A legfontosabb, jövőbe mutató előnye, viszont

a mesterséges intelligencia terén való kifejezetten jó és nagy népszerűségnek örvendő alkalmazhatósága.

A termékelőállítás tervezési szakaszában a gyártási idők teljes pontosságú meghatározása nem mindig végezhető el gazdaságosan, ugyanakkor az egyes fázisok esetében nincs is szükség rá. Megfelelő pontossággal bíró becslés adatai lehetővé teszik a tervezési folyamatok hatékony elvégzését. A gyártási idő pedig többféleképpen felhasználható, legyen szó ütemezésről vagy költségbecslésről, mi az utóbbival szeretnénk foglalkozni termék gyártási költséget szeretnénk becsülni. A program a terméktervezési fázisban is segítséget nyújthat a különböző koncepciók költségvonzatainak összehasonlításával, így több iterációs lépéstől megkímélheti a tervezésben és a gyártásban résztvevőket. A kinyert adatok továbbá bemenő adatként szolgálhatnak egy magasabb rendű gyártásütemezési szimuláció elvégzéséhez, amivel a szimuláció optimalizálásához szükséges idő lerövidíthető. Mindemellett egy, a programban jól felépített és aktualizált adatbázis segítségével, akár a koncepció megszületésekor vizsgálható a szükséges humán- és tárgyi erőforrás kapacitásának rendelkezésre állása.

Összefoglalva, a dolgozat egy általunk fejlesztett szoftver működését és felépítését hivatott bemutatni, amely egy STEP fájl beolvasásából várható gyártási időt és gyártási költséget biztosít a felhasználó számára.

II A STEP fájlok alapja

II.1 Mi is a STEP

A CAD szoftverekből, például a CATIA-ból, Creo-ból vagy SolidWorks-ből származó natív CAD -fájlformátumok saját felépítéssel rendelkeznek, nem fordíthatók le közvetlenül, és nem nyithatóak meg más szabadalmaztatott CAD -rendszerrel. Azonban azáltal, hogy a CAD rendszerek képesek semleges formátumba exportálni, mint például a STEP, lehetővé válik az interoperabilitás, azaz átjárás a különböző gyártók között. Ennél is fontosabb, hogy mivel a vállalatok a modellalapú információátvitel irányába mozdulnak el, és eltávolodnak a hagyományos papír alapú munkafolyamatoktól, a STEP, a legelterjedtebb fájlformátum a különböző munkacsoportok, beszállítók, alvállalkozók és ügyfelek közötti kooperációban.

Ez más fájlformátumokkal ellentétben egy jól szabványosított és rengeteg különböző plusz gyártási információt magában hordozó megoldás. A “Standard for Exchange of Product model data”, röviden STEP, EXPRESS modellezési nyelven íródott, melyet az ISO-10303 szabvány rögzít.

A STEP formátum létrehozása az 1980-as évekre datálható, elsődleges célja a szabványosítás volt, így biztosítva a különböző CAD rendszerek közötti átjárást. Ez az átjárás lényegében a fájlok kompatibilitását jelenti, más-más szoftverekkel, úgy, hogy az elkészített modellek szoftver függetlenül megoszthatók és módosíthatók legyenek. A platformok közötti kompatibilitás mellett a STEP-fájlok nagy előnye a pontosság, különösen mondjuk az STL-hez hasonló fájlokhoz képest.

Először is, ellentétben az STL fájlokkal, amelyek csak a modell hálós külső geometriáját írják le, a STEP fájl nem bontja a felületeket véges számú elemek sokaságára, a leírás egyértelműsíti a kirajzolandó geometria alakját és helyzetét, azonban a kirajzolás felbontását a beolvasó szoftverre bízta. A hálózott geometriák hátránya, hogy le vannak egyszerűsítve, az STL esetében háromszögelésről beszélhetünk, amelyek összessége a test körvonalát alkotja. A STEP fájlokban ezzel ellentétben minden különálló felület egyetlen entitásként van eltárolva, így a fájlformátumot ideálissá téve nagy részletességet igénylő alkalmazásokban. A különbséget egyértelműen mutatja az a 2. Ábra. Tehát, amennyiben részletes 3D modellekre van szükségünk, például megmunkáláshoz vagy 3D nyomtatáshoz a STEP fájlformátum az egyik, ha nem a legjobb választás.[1.]

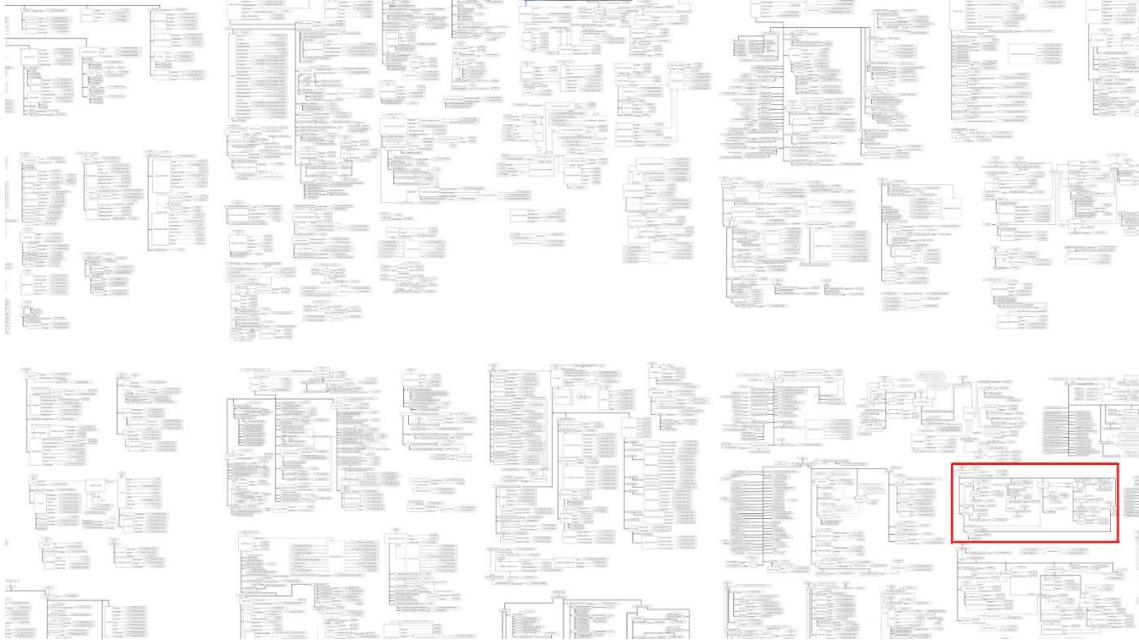


2. Ábra: STL és STEP[1.]

A STEP eredetileg egyetlen, teljes, megvalósítástól független termékinformációs modellre épül, amely az integrált helyi és alkalmazási információs modellek törzsrekordja, de bonyolultság miatt a szabványt kisebb részekre kellett bontani. Felhasználási területtől függően többféle alkalmazás protokoll (AP) létezik. N. Jimmy [2.] behatóan foglalkozott a témával és sorba szedte a legelterjedtebb AP-kat. Az AP203 csak mechanikai alkatrészek és szerelvények 3D konfigurációját írja le, míg az AP214 már alap adatokat tartalmaz az autóipari gépészeti tervezési folyamatokhoz (pl.: geometriai méretezések és tűrések, színek, kinematikai információk, szöveges megjegyzések...). Ezzel szemben az AP242 egyesíti mindkét protokoll előnyeit, valamint tartalmazza a PMI-t (termék gyártási információk), például a GD&T-t, a BOM-ot és más információkat az alkatrész vagy összeszerelés elkészítéséhez és méréséhez egy egységes fájlformátumban, így biztosítva különböző tervezési rendszerek (CAD -CAD vagy CAD -CAE) közötti átjárhatóságot, valamint gyártás esetében a termékdefiníció átvitelét a gyártási/mérési szoftverbe (CAD -CAM vagy CAD -CMM).

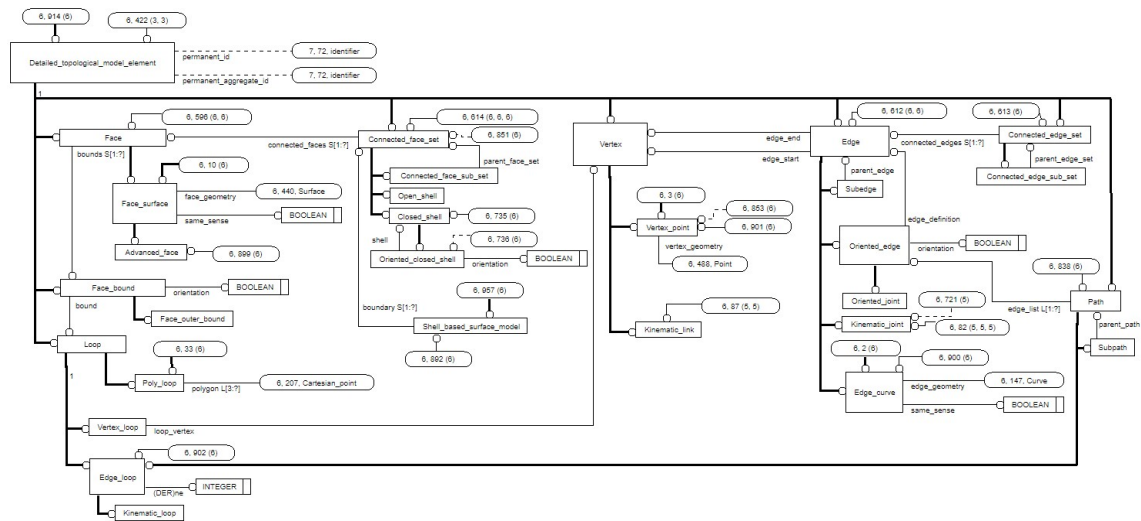
II.2 A STEP fájlok felépítése

A STEP fájlok felépítése igen bonyolult és szerteágazó. Az előző fejezetben is kitértünk arra, hogy miért is kellett AP-kra bontani a teljesen egységes formátumot, most azonban ebből a hatalmas adatmennyiségből csak egy kis szeletet szeretnénk részletesen tárgyalni. Viszonyítási alapként a formátum térképén bejelöltük az általunk használt kizárólag CAD geometriákra vonatkozó részt, lásd vörössel keretezve a 3. Ábrán.



3. Ábra: Egy komplex AP felépítése [3.]

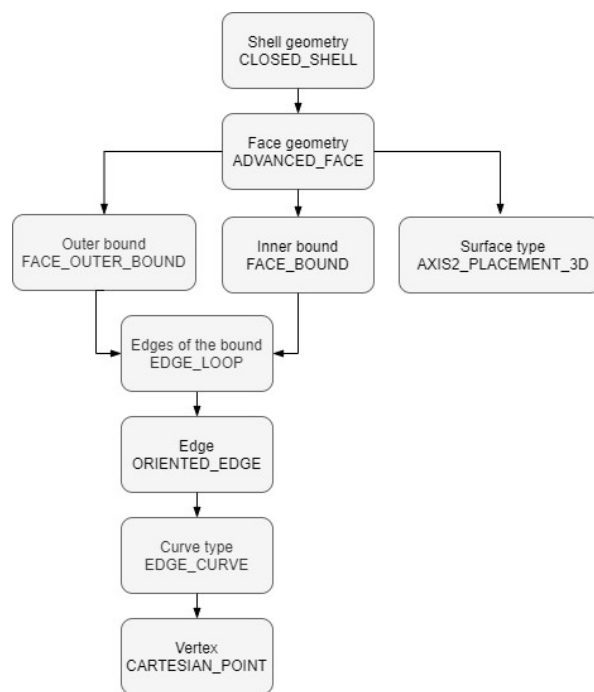
A térkép közlése a teljesség és szemléltetés kedvéért szükséges, ugyanis a különböző kapcsolatok ábrázolása ezen a legpontosabb. Itt nem csak az entitások és azok kapcsolatai vannak ábrázolva, hanem már azok attribútumai és azok kapcsolatai is. Érezhető, hogy a felépítési struktúra ábrázolása és megértése ebből az ábrából nehézkes. A következőkben a STEP fájl számunkra fontos a 3. Ábrán vörössel bekeretezett részegységének bemutatásával és elemzésével foglalkozunk. (lásd 4. Ábra)



4. Ábra: A számunkra szükséges adathalmaz [3.]

Maga a STEP fájl tulajdonképpen egy szöveges állomány, amely egy munkadarab geometriai adatait tartalmazza határaival, valamint ábrázolási adataival, úgy, mint héjak, oldalak, élhurkok, csúcsok; felületi geometriai adatokkal, mint pl. síkok, hengerek, kúpok, gyűrűk, gömbök; görbe geometriákkal, például vonalak, körök, ellipszisek. Ahogy az a fentebbi ábrán is látható minden egyes felsorolt geometriai építőelem rendelkezik a modell rekonstruálásához szükséges attribútumokkal. Ezek a strukturált szövegben a különböző rekordok függvényattribútumaiként jelennek meg.

A STEP fájlok határolófelület-alapú, úgynevezett B-rep felépítéssel rendelkeznek, ami azt jelenti, hogy a határoló elemek topológiai sorrendben vannak eltárolva úgy, hogy a köztük lévő kapcsolatokról és szabályszerűségekből épüljön fel a teljes geometria. A belső szilárd anyagmennyiséget zárt héj határozza meg. Zárt héj felületekből áll. Minden egyes felületet határoló vonalak szegélyeznek, ezek élhurokból épülnek fel. Egy élhurok orientált éleket tartalmaz. Minden él két csúcs között húzódik, amelyek koordinátái kinyerhetők. Más szóval, ha fel szeretnénk dolgozni egy STEP fájlt, akkor a modelleket a legkisebb egységig le kell bontani, ott információt kinyerni, majd az adatokkal az adott topológiának megfelelően rekonstruálni. Bhandarkar és Nagi [4.] ezt a topológiai felépítést egy fa struktúrában ábrázolta (5. Ábra).

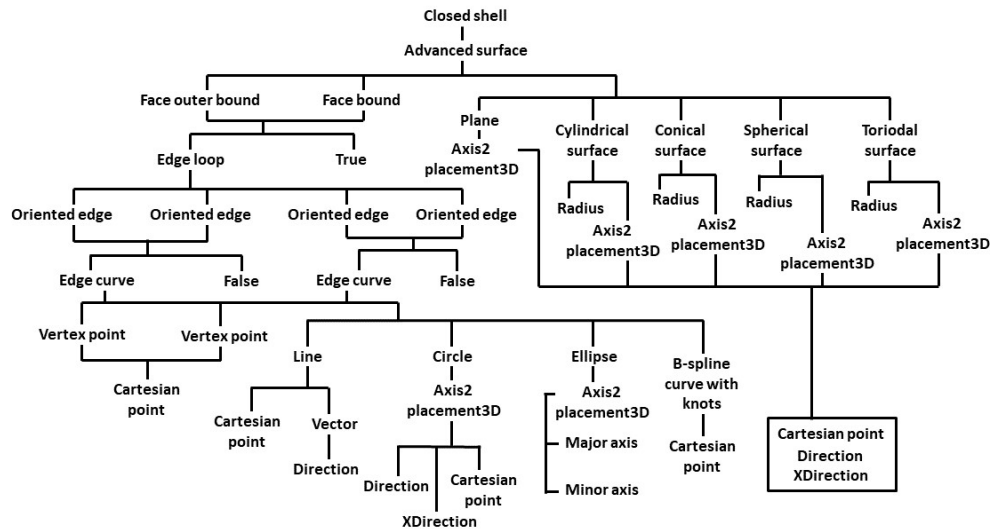


5. Ábra: A hierarchikus topológia

A térkép attribútumai a következők [5.]:

- Zárt héj (CLOSED_SHELL): Több felület gyűjteménye, amely háromdimenziós térben határol egy területet, és két részre osztja a teret, az egyik véges, a másik végtelen.
- Felület (ADVANCED_FACE): Felülettípus, amelyben a geometriát a hozzá tartozó orientációs adatok, külső és belső határoló görbék írják le.
- Élhurok (EDGE-LOOP): Élek olyan útvonala, amelyben a kezdő és a végpont azonos.
- Orientált él (ORIENTED_EDGE): Két adott pont között húzódó szakasz, amely tartalmazza annak irányát. Az ORIENTED-EDGE egyenértékű lesz az eredeti éllel, amennyiben attribútuma alapján orientációját nem kell negálni.
- Élgörbe (EDGE_CURVE): Olyan éltípus, amelynek geometriája teljesen meghatározott, például egyenes, körív, B-Spline.
- Csúcspont (CARTESIAN_POINT): A csúcs geometriáját meghatározó pont koordinátái Descartes koordináta rendszerben.

A fa struktúrát pedig teljesen kibontva, részletesen Malleswari [6.] ábrázolta.



6. Ábra: A részletes fa struktúra

Az 6. Ábra felépítésében és a tartalmazott információk mennyiségében is elég letisztult ahhoz, hogy segítségével a STEP fájlok megértését elősegítsük. Az ábrán látható az adott entitásokhoz tartozó néhány geometriai attribútum is, amelyek kinyerésével és felhasználásával a következő fejezetekben foglalkozunk.

Maga a fájl egy magasan strukturált szövegfórmátum, mely a kódsorok egymásra hivatkozásával teremti meg a hierarchikus kapcsolatrendszert (7. Ábra). Az ábrán jól látható, hogy a rekordok azonosítói a sor elején hashmark-al kezdődnek és egy a sorszámuknak megfelelő szám azonosítja őket. A különböző függvényattribútumok pedig tartalmazzák a hivatkozott rekordok azonosítóit, konkrét adatokat, valamint az entitás megnevezését. A STEP fájlt teljes egészében ilyen, és ezekhez hasonló sorok alkotják.

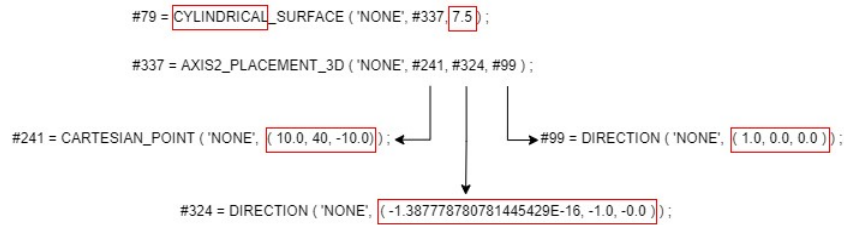
```
#15=FACE_BOUND(' ',#35,.T.);
#16=FACE_BOUND(' ',#41,.T.);
...
#24=FACE_OUTER_BOUND(' ',#32,.T.);
...
#32=EDGE_LOOP(' ',(#101,#102,#103,#104));
#33=EDGE_LOOP(' ',(#105,#106,#107,#108));
...
#42=LINE(' ',#218,#56);
...
#70=CIRCLE(' ',#165,3.5);
#72=CIRCLE(' ',#168,2.5);
...
#83=VERTEX_POINT(' ',#244);
...
#98=EDGE_CURVE(' ',#82,#83,#53,.T.);
#99=EDGE_CURVE(' ',#83,#77,#54,.T.);
...
#132=ORIENTED_EDGE(' ',*,*,#96,.T.);
...
#140=ADVANCED_FACE(' ',(#28),#19,.T.);
#141=ADVANCED_FACE(' ',(#29),#20,.T.);
...
#144=CLOSED_SHELL(' ',(#136,#137,#138,#139,#140,#141,#142,#143));
```

7. Ábra: A STEP fájlok szöveges felépítése

II.3 A STEP fájlok feldolgozása

Mint már említettük, az a rekonstrukcióhoz szükséges legtöbb információ a hierarchikus struktúra legalján helyezkednek el, amihez a teljes fájl feldolgozásán keresztül juthatunk el. A STEP fájl feldolgozása CLOSED_SHELL karakterlánc megkeresésével kezdődik, és a szükséges koordinátákkal végződik. Ezek kinyeréséhez reguláris kifejezések beágyazott ciklusai szükségesek. A feldolgozáshoz használt algoritmus fontos, hogy kövesse a stringben meghatározott topológiai sorrendet, így lehetővé téve a szükséges információk kinyerését a munkadarab külső határoló burkától az összes sarokpontig.

A különböző szinteken kinyert információkat a 8. Ábra mutatja. Fontos észrevenni, hogy a különböző rekordok azonosítói nagyon eltérőek, így már ez alapján belátható, hogy a rekordok sorrendje teljesen tetszőleges. Ez a kijelentés a szekvenciális feldolgozást teljesen kizárja.



8. Ábra: Az egymásba ágyazott struktúra

A legtöbb entitás rendelkezik plusz attribútumokkal az öt alkotó elemeken kívül. Ezek lehetnek konkrét értékek, valamint sok esetben akár egy Boolean érték, amelyeknek orientációs és összekapcsolási szempontból van szerepük.

A dolgozat egyik fő tématerülete a STEP fájlok feldolgozása. Az általunk használt feldolgozási módszert, algoritmust és azok leírását később részletezzük, azonban alapvető szabályokról fontos, hogy már most ejtsünk néhány szót. Előljáróban annyit, hogy vannak olyan törvényszerűségek, amelyek a legtöbb fellelhető szakirodalomban megtalálhatóak, valamint gyakorlatban is teszteltük őket, így tehát biztos alapot szolgáltatnak számunkra az algoritmusunk fejlesztéséhez. A legfontosabbak a következők:

- az adott rekordok sorai mindig ‘ # ’ karakterrel kezdődnek
- a formátum nem tiltja a sorokban, illetve sorok között elhelyezett whitespace karaktereket
- egy rekord lezárása ‘ ; ’ karakterrel történik
- az attribútumok közötti határolójel nyelvtérülettől függetlenül ‘ , ’ (vessző)
- a különböző rekordok függvényattribútumainak sorrendje szabványban rögzített
- a felületek normálvektorai mindig a felületből kifelé néznek
- a vektorok irányítottságát a végpontjaik felsorolásának sorrendje definiálja
- a definiált irányt a Boolean attribútumok FALSE értéke negálja
- az irányított élek esetén a negálás értelme, hogy a viszonyított felületen az élek mindig zárt láncot alkossanak
- az ADVANCED_FACE entitás Boolean attribútuma a felület normálisának negálására szolgál, itt a fentebb említett normál vektorokra vonatkozó konvenció betartása a cél
- a felületeket alkotó élek mindig olyan irányítottságúak, hogy az anyag tőlük balra helyezkedjen el
- két szomszédos sík normálvektorának keresztszorzatából kapott eredményvektor segítségével a felületek közötti konkáv/ konvex kapcsolat determinálható

III Alaksajátosságok

III.1 Mi is az a feature

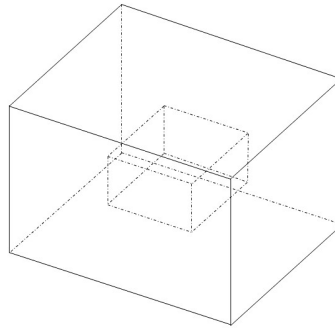
Amikor 3D-s geometriákról beszélünk a feature-ök, azaz alaksajátosságok többféle módon definiálhatók, van a felhasználási területekhez kötött definíció [7.], és van az a meghatározás, mikor folyamattervezés irányából közelítik meg a témát [8.]. A mi értelmezésünkben az alaksajátosságok az alkatrész pontjainak, éleinek, felületeinek, olyan logikai összerendelését jelentik, amely célzottan befolyásolja az alkatrész alakját. A legfontosabb értelemszerűen a magasabb rendű, a gyártástervezés tekintetében hangsúlyos alaksajátosságok megkeresése, amelyen furatok, zsebek és letörések is elhelyezkedhetnek. Azonban a STEP fájlokban ezek alsóbbrendű alaksajátosságokkal vannak definiálva, ezek a 2 és 3D-s entitások, amelyek hierarchikus felépítését az előző fejezetben tárgyaltuk. Srivastava és Komma [9.] a munkájukban már komplexebb szempontból körbejárták a témát, előállítottak egy termékadatmodellt, amely olyan információkat tartalmaz az alaksajátosságokról, amelyeket a CAM és CAPP rendszerek használnak. Ezek geometriák, toleranciák, gyártási és NC információk. Ők a magasabb rendű feature-öket már gyártási alaksajátosságnak nevezték.

Az alaksajátosságok csoportosításának mérnöki szemlélete szerint a gyártási alaksajátosságok azaz „form features” egy a terméktervezés során létrejövő egymással kapcsolt felületek sorozata, amelyek létrehozása a megmunkálás célja. Ezeket behatóan vizsgálva a megmunkálási műveletek típusa meghatározható, ilyenek például a fúrás, esztergálás és marás. [10.]

Az osztályozás alapja az alaksajátosságok hatása a kiindulási geometriára. A négy fő kategória [4.]:

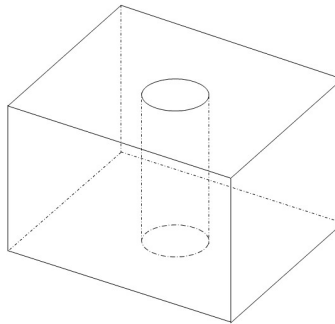
1. A test belsejében lévő alaksajátosságok (Void form features)
2. Belső formai alaksajátosság (Internal form features)
3. Külső formai alaksajátosság (External form features)
4. Összeköttetéseket módosító alaksajátosságok. (Connectivity modifying form features)

A test belsejében lévő alaksajátosságokhoz tartozó alak jellemzők üregeket képeznek a szilárd testben, ezek olyan térfogatok, amelyek teljesen be vannak zárva az anyagba.



9. Ábra: Test belsejében lévő alaksajátosság

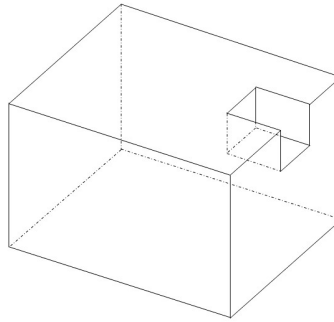
Belső formai alaksajátosságok azok, amelyek megváltoztatják egy, vagy több felület belső alakját. Két alcsoportba soroljuk. Átmenőnek nevezzük azokat, amelyek a munkadarabon keresztül utat képeznek két külső felület között, ide tartoznak az átmenő furatok és prizmatikus üregek. Zsák kategóriába soroljuk azon felületek belsején megjelenő alaksajátosságokat, amelyek behatolnak az anyagba, de másik felületen nem lépnek ki belőle, ezek a zsákfuratok és zsebek.



10. Ábra: Belső formai alaksajátosság

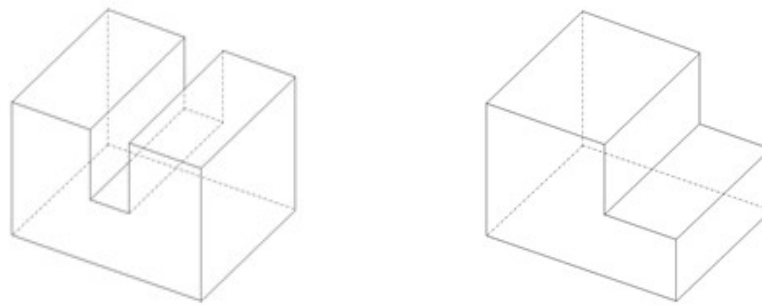
A külső formai alaksajátossághoz tartozó jellemzők módosítják a felület külső alakját, ezek például a lépcsők, külső zsebek és hornyok. A külső formai alaksajátosságokat további három altípusba sorolhatjuk:

- Élmódosítók: ezek módosítják a felület egyik élét és a felületepárok között alaksajátosságot hoznak létre. Pl: külső zseb.



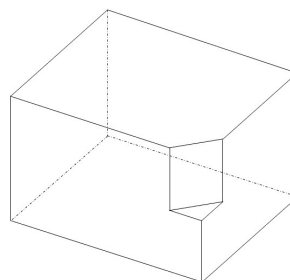
11. Ábra: Élmódosító alaksajátosság

- Felületmódosítók: itt egy felület kerül módosításra úgy, hogy a teljes felület szétválasztásával egy alaksajátosság jelenik meg rajta. A mérnöki alkalmazásokban ezek a leggyakrabban előforduló alaksajátosságok. Például hornyok és lépcsők.



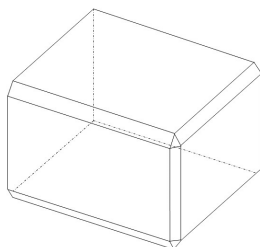
12. Ábra: Felületmódosító alaksajátosság

- Csúcsmódosítók: ezen alaksajátosságok módosítják a szomszédos élek közötti csúcsokat.



13. Ábra: Csúcsmódosító alaksajátosság

Összeköttetéseket módosító alaksajátosságok jellemzői, hogy megváltoztatják a kapcsolatot két felület között, például letörések és lekerekítések. Ezeket jellemzően utolsóként hozzák létre a gyártási folyamat során.



14. Ábra: Élmódosító alaksajátosság

III.2 Irodalmi áttekintés az alaksajátosságok felismerésének témakörében

A 3D-s munkadarabok alaksajátosság felismerésén alapuló feldolgozása egy igen népszerű kutatási terület. Az irodalom több módszert is taglal, ami ezt a problémakört boncolgatja [5.][11.]. Itt érdemes előre leszögezni, hogy a legtöbb esetben ezeknek a módszereknek a leírása csakis elméleti síkon történik, megvalósíthatóságát, valamint megvalósítását a legtöbb esetben csak részlegesen vagy egyáltalán nem tárgyalja az irodalom.

Gráf alapú módszerek

A gráfokon alapuló megközelítésben a szilárd modell B-reprezentációját használják egy szomszédsági gráf (AAG) elkészítéséhez, amelynek vizsgálatával az alaksajátosságok felismerhetők. A B-reprezentáció a már bemutatott térképen ábrázolja a topológiai elemek (felületek, élek és csúcsok) hierarchikus felépítését. A modell minden eleméhez geometriai tulajdonságok is tartoznak, amelyekből a felületek, élek és csúcsok szomszédsági információi kinyerhetők, ezek elengedhetetlenek az alaksajátosságok felismeréséhez. A feldolgozás az AAG segítségével történik, melynek csomópontjai a felületek és az őket összekötő egyenesek a test élével ekvivalensek. Az összeköttetések egyedi attribútummal rendelkeznek, amely a két felület egymáshoz viszonyított konvex vagy konkáv kapcsolatát jelöli. A módszer szabályszerűségek és minták segítségével határoz meg síkbeli alaksajátosságokat.[12.]

Egy alternatív módszer az él-felület gráf (GEFG) elkészítése és vizsgálata, az AAG -val ellentétben a GEFG további két topológiai entitást használ (a héjat és a hurkot) a gráf

felépítéséhez. A héj a csatlakoztatott felületek maximális számát reprezentálja, míg a hurok a záródó éleket. A módszer lebontja a GEFG-t két-és háromkapcsolt algráfokra. A módszer itt is szabályszerűségeken és mintákon alapul, azonban itt hengeres elemeket is képesek vagyunk azonosítani.[13.]

A gráfokon alapuló megközelítés jól alkalmazható szögletes alaksajátosságok felismerésére, valamint hengeres felületek is megtalálhatóak, de ezekhez bővíteni kell a gráfot további attribútumokkal, a végeredmény pedig így túlbonyolított és nagy számításigényű lehet.

Intuíció-alapú módszer

Az intuíció-alapú megközelítés logikai szabályokat használ egy szilárdtest topológiai adatai alapján az alaksajátosság felismerésére vonatkozó “megérzések” létrehozásához. A szilárd modell felületeit, topológiai adatait és geometriai információit vizsgálja. Ezek a kvázi tippek csak részleges ábrázolást képeznek az alaksajátosságokról, amely még további elemzést igényelnek a teljes megismeréshez. Az intuíció-alapú megközelítés fokozatosan vizsgálja az alaksajátosságok lehetséges építőelemeit, míg a gráfokon alapuló megközelítés egyszerre határozza meg őket. Például az esetek nagy részében a hengeres felületek utalhatnak a furatok jelenlétére, míg a párhuzamos oldalak az anyag belsejében összekötve zsebek lehetnek.[14.]

Intuíciók használatával generálhatunk egyéb információkat, például szemantikát és geometriai attribútumokat az alkatrészből. Az intuíció alapú alaksajátosságfelismerő rendszer általában egy előre letárolt készlettel rendelkezik a lehetséges alakzatokról és ezeket próbálja az adott alkatrésze alkalmazni. Az alapelv az, hogy ha egy alkatrész az a kiindulási geometria megmunkálásával előállítható, akkor belőle eltávolított térfogat az alkatrész alaksajátosságait jellemzi. [14.]

Összefoglalva, az intuíciókra alapozott megközelítést túlnyomórészt az általános alaksajátosságok felismerésére használják, ezek olyan megmunkálási jellemzők, amelyek fúrásból, marásból, letörésből és lekerekítésből alakultak ki és előre meghatározott könyvtárban vannak tárolva. Az intuíciókra épülő megközelítés speciális szabályokat alkalmazva generálja, osztályozza és veti el/választja ki a “megérzéseket”.

A konvex burok módszer

A konvex burok módszer a teljes szilárd geometriát használja, úgy, hogy a komplex struktúrát primitívekre bontja. Az első lépésben létrehozásra kerül a munkadarab befoglaló méreteiből készíthető legkisebb nem konkáv szögletes burok, majd ebből a munkadarabot kvázi kivonva

előáll az úgynevezett negatív térfogat, amelyből az alaksajátosságok kinyerhetőek. Hengeres felületek vizsgálatára alapvetően nem alkalmas.[15.]

Egység alapú térfogat bontási módszer

Az egységekre bontás egy adott alakzat térfogati cellákra való bontását jelenti úgy, hogy az egységek újra egyesítése adja az eredeti alakzatot. A konvex burok módszerrel ellentétben itt a negatív térfogatot bontjuk egységnyi térfogatú cellákra a burok használata nélkül. Ezt követően az egységnyi térfogatú cellákat kombinálva előáll a maximális térfogat, amely egy adott alaksajátossággal ekvivalens. Az egységnyi térfogati bontási módszer három lépésből áll. Először az alkatrészt kivonják a kiindulási nyersanyagból, hogy a negatív térfogat előálljon. A negatív térfogat egységcellákra bontható a kiválasztott felületek vagy légrések segítségével. Másodszor, az egységcellákat egyesítjük, hogy maximális térfogatokat hozzunk létre a gyártási műveletekhez kapcsolódó korlátozások figyelembevételével. Végül az utolsó lépésben a maximális térfogatok osztályozása történik meg, így megállapítva az alaksajátosságokat, mint speciális megmunkálási jellemzők. Az újraalkotás algoritmus az egységcellák maximális térfogatába rendezésének szempontjából drága, mert nem mindig egyezik az előre definiált maximális térfogatokkal. A megközelítés alkalmas a szögletes alaksajátosságok felismerésére, de a problémák továbbra is fennállnak a szabad formájú és ívelt felületekkel.[16.]

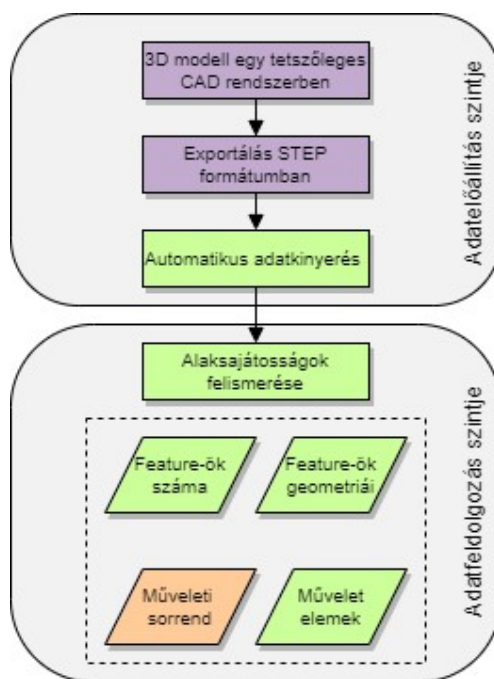
Hibrid megközelítés

A hibrid megközelítés a korábban tárgyalt stratégiák kombinációját használja, leküzdve az egyes módszerekben továbbra is fennálló korlátokat, többnyire alaksajátosságok egymással való metszésére vonatkoztatva.

A gráfokon- és szabályszerűség-alapú megközelítések kombinációjából előáll egy olyan algoritmus, amely képes hatékonyan alacsony számításigénnyel felismerni az egymást metsző alaksajátosságokat. Ez az algoritmus kiterjesztett hozzárendelt szomszédsági gráfot (EAAG) használ, amely az AAG-n felül további csomópontokat és éleket alkalmaz, így szétválasztva az egymást keresztező sajátosságokat.[17.]

Rameshbabu és Smunmugan [18.] definiált egy hibrid alaksajátosság felismerő metódust, amely egy szomszédsági gráfot és negatív térfogatot használ. A szerzői a [19.] cikknek metódusokat kerestek ún. szabad alaksajátosságok felismerésére gyártásanalízis segítségével. Bhandarkar és Nagi [4.] STEP alapú alaksajátosság kicsomagolási algoritmust fejlesztettek, amely prizmatikus munkadarabok B reprezentációját egyszerűsíti.

IV Az elkészült program



15. Ábra: 3D-s modell feldolgozási egység

A következőkben a 15. Ábrán látható Adatelőállítás és Adatfeldolgozás szintjével fogunk behatóan foglalkozni, elméleti és gyakorlati oldalról is megvizsgáljuk őket. Majd a fejezet végén bemutatásra kerül az elkészült program és felhasználói felülete.

IV.1 STEP fájl feldolgozás

IV.1.1 Felhasznált elméleti alapok

A megmunkálási területeken és alapvetően az informatikában is igen gyakori, hogy szövegfájlban tárolunk információt, elég csak például az NC nyelvre gondolni. A kulcs itt, hogy általános megfeleltetéseket használunk, amelyet a feldolgozóegység is ismer, így tehát a kívánt információk a megfelelő algoritmus segítségével kinyerhetők. Jeffrey [20.] behatóan foglalkozott a témával és a reguláris kifejezések feldolgozásával, ő úgy definiálta a problémát, hogy képesek legyünk feldolgozni a szövegfájlokat, olyan karakterláncokat kell keresnünk, amely képes a lényegi információk szétválasztására.

A strukturált szövegfájlok beolvasását a reguláris kifejezésekkel dolgozó parse modulok végzik. A beolvasás folyamán minta kereséssel találjuk meg az értelmezhető részegységeket, melyek az úgynevezett tokenek, majd meghatározzuk az egyes tokenek közötti relációkat. Az értelmezhető szöveges egységek lesznek a reguláris kifejezések, az adott mondat nyelvtanát

pedig a tokenekből épített struktúra szemlélteti. Ahogy, a 16. Ábrán is látható, a reguláris kifejezések olyan minták, amelyek képesek leírni a szavakat, speciális jelentéssel bíró karakterei pedig a metakarakterek. A mintakeresés kulcsa ezen metakarakterek, valamint az általuk megjelölt információhalmaz típusának ismerete. Ez azt jelenti, hogy nem tudjuk pontosan mi lesz az, amit egy adott metakarakter megjelöl, de tudjuk, hogy milyen részegységek építhetők fel, karakterek, számok, írásjelek és ezek tetszőleges kombinációja. Az általunk használt mintákat, tehát ennek megfelelően, univerzálisan kell elkészíteni.

RegularExpression["\\s[+-]?[0-9]*\\. [0-9]*\\s"]

A fenti ábrán egy valós számok megtalálására használt minta látható. A valós számok két részből állnak egész rész és törtész. Mindkét rész 0-9 karakterek tetszőleges kombinációja, melyeket a tizedespont választ el. A szám előtt [+ -] előjel lehetséges, de nem kötelező, ezt hivatott jelölni a ?, míg a * a számosságra utal (0, 1, több). A \\s white space karakterek a számokat választják el egymástól, míg \. a határoló karaktert jelenti a szám egész és tört része között.

16. Ábra: Reguláris kifejezés

IV.1.2 Megvalósítás

Gyakorlatban elsőnek meg kell állapítani a metakaraktereket, amelyeket keresünk, valamint az elválasztó elemeket. Jelen esetben metakifejezésekkel van dolgunk, amelyeket az irodalomból, illetve néhány fájl vizsgálata után magunktól is felismerhetünk. A STEP feldolgozása tehát ezen metakarakterek által megjelölt információ kinyerésével ekvivalens. Esetünkben többszintes, egymásra hivatkozásokkal tarkított struktúráról beszélhetünk. Lényegében a már bemutatott térképet kell követnünk és leásni egészen a testet meghatározó csúcspontokig. A kialakított struktúra programozástechnikai megvalósítását a 17. Ábrán szemléltetjük.

```
p = re.compile(r'#\d*=CLOSED_SHELL[^;]*;')
CLOSED_SHELL = p.findall(string)

["#245=CLOSED_SHELL('NONE', (#367,#156,#224,#427,#24,#89,#290,#4,#33,#429,#76,#384,#407,#337,#514,#250,#150,#453,#409,#284));"]

p = re.compile(r'#\d*')
ADVANCED_SURFACE_NUMBER = p.findall(CLOSED_SHELL[0])[1:]

['#367', '#156', '#224', '#427', '#24', '#89', '#290', '#4', '#33', '#429', '#76', '#384', '#407', '#337', '#514', '#250', '#150', '#453', '#409', '#284']

for i in ADVANCED_SURFACE_NUMBER:
    p = re.compile(i + r'=[^;]*;')
    ad_surf = p.findall(string)[0]
    ADVANCED_SURFACE.append(ad_surf)

["#367=ADVANCED_FACE('NONE', (#489),#49,.T.);", "#156=ADVANCED_FACE('NONE', (#261),#459,.F.);", "#224=ADVANCED_FACE('NONE', (#109),#366,.F.);", "#427=ADVANCED_FACE('NONE', (#344),#179),#241,.F.);", "#24=ADVANCED_FACE('NONE', (#98),#196),#387,.T.);", "#89=ADVANCED_FACE('NONE', (#69),#603,.F.);", "#290=ADVANCED_FACE('NONE', (#317),#266,.T.);", "#4=ADVANCED_FACE('NONE', (#87),#535,.T.);", "#33=ADVANCED_FACE('NONE', (#96),#336,.T.);", "#429=ADVANCED_FACE('NONE', (#437),#539,.F.);", "#76=ADVANCED_FACE('NONE', (#589),#143,.F.);", "#384=ADVANCED_FACE('NONE', (#391),#46,.F.);", "#407=ADVANCED_FACE('NONE', (#528),#536,.F.);", "#337=ADVANCED_FACE('NONE', (#468),#420,.F.);", "#514=ADVANCED_FACE('NONE', (#451),#108,.F.);", "#250=ADVANCED_FACE('NONE', (#556),#115,.T.);", "#150=ADVANCED_FACE('NONE', (#309),#351,.F.);", "#453=ADVANCED_FACE('NONE', (#335),#534,.T.);", "#409=ADVANCED_FACE('NONE', (#95),#625,.T.);", "#284=ADVANCED_FACE('NONE', (#117),#82,.T.);"]
```

17. Ábra: Program minta

Az ábra célszerűen a kód elejéről a zárt burok megtalálását és abból az elkülönülő felületek kinyerését mutatja be. Azért célszerű ezt a részt elővenni, mivel itt még csak egy dimenziós listákkal dolgozik a program és egyszerű átlátni annak felépítését. Az első kódrészlet értelemszerűen kizárólag a CLOSED_SHELL szöveget tartalmazó rekordot keresi. Ezt a #245-ös azonosítónál meg is találta. Követve a térkép útmutatását tudhatjuk, hogy ezen sor attribútumai az ADVANCED_FACE rekordjaira mutatnak. Először kinyerjük az azonosítókat, majd azok segítségével az ADVANCED_FACE-ek teljes rekordjait is. Látható, hogy az ADVANCED_FACE attribútumai között már több különböző entitás is szerepel ezért komplexitása igen, viszont kinyerésének és tárolásának logikája nem változik.

Az információkat a metakarakterek, valamint a szükségtelen elemek nélkül listákban tudjuk eltárolni. A programunk a STEP feldolgozása után előállít több adatbázist is, amelyek a lényegi nyers információkat tartalmazzák, ezek a felhasználói felületen is megtekinthetők, lásd 18. Ábra, illetve 2. függelék.

ID	Surface	Radius	Half cone angle	Adv.Surf. boolean	Position	Normal Vector	Reference Direction
1	CYLINDRICAL	7.5		T	[1.5959455978986622e-14, 25.0, -39.99999999999998]	[-1.3877787807814457e-16, -0.0, 1.0]	[1.0, 0.0, 0.0]
2	CYLINDRICAL	4.488075498384091		F	[15.0000000000000021, 40.000000000000036, -15.000000000000014]	[-0.0, 1.0, -0.0]	[0.0, 0.0, 1.0]
3	PLANE			F	[-25.0, 50.0, 24.99999999999999]	[1.3877787807814454e-16, 0.0, -1.0]	[-1.0, 0.0, -1.3877787807814454e-16]
4	PLANE			F	[0.0, 50.0, 0.0]	[0.0, -1.0, 0.0]	[0.0, -0.0, -1.0]
5	PLANE			T	[-24.999999999999986, 50.0, -25.000000000000014]	[1.3877787807814454e-16, 0.0, -1.0]	[-1.0, 0.0, -1.3877787807814454e-16]
6	PLANE			F	[-24.999999999999986, 50.0, -25.000000000000014]	[1.0, -0.0, 2.7755575615628914e-16]	[2.7755575615628914e-16, 0.0, -1.0]
7	PLANE			T	[25.000000000000002, 50.0, -25.000000000000007]	[1.0, 0.0, 0.0]	[0.0, 0.0, -1.0]
8	PLANE			T	[0.0, 0.0, 0.0]	[0.0, -1.0, 0.0]	[0.0, -0.0, -1.0]
9	PLANE			T	[-10.000000000000002, 50.0, -4.999999999999999]	[-1.0, -3.469446951953616e-16, 0.0]	[3.469446951953616e-16, -1.0, 0.0]
10	PLANE			F	[-15.000000000000002, 45.000000000000001, -4.999999999999999]	[0.0, -1.0, 0.0]	[0.0, -0.0, -1.0]
11	PLANE			F	[-15.000000000000002, 50.0, -4.999999999999999]	[-1.0, 0.0, -0.0]	[-0.0, 0.0, 1.0]
12	PLANE			F	[0.0, 0.0, -4.999999999999999]	[0.0, 0.0, -1.0]	[-1.0, 0.0, -0.0]
13	PLANE			F	[15.0000000000000016, 46.000000000000004, 14.999999999999995]	[-1.0, 0.0, 0.0]	[0.0, 0.0, 1.0]
14	CYLINDRICAL	2.5000000000000004		F	[17.5000000000000018, 46.000000000000004, 14.999999999999995]	[-0.0, 1.0, -0.0]	[0.0, 0.0, 1.0]
15	PLANE			F	[20.000000000000002, 46.000000000000004, 24.999999999999993]	[1.0, 0.0, 0.0]	[0.0, 0.0, -1.0]
16	PLANE			T	[17.5000000000000018, 46.000000000000004, 14.999999999999995]	[0.0, 1.0, 0.0]	[0.0, -0.0, 1.0]
17	CYLINDRICAL	4.488075498384091		F	[15.0000000000000021, 40.000000000000036, -15.000000000000014]	[-0.0, 1.0, -0.0]	[0.0, 0.0, 1.0]
18	PLANE			T	[15.0000000000000021, 40.000000000000036, -15.000000000000014]	[0.0, 1.0, 0.0]	[0.0, -0.0, 1.0]
19	CYLINDRICAL	7.5		T	[1.5959455978986622e-14, 25.0, -39.99999999999998]	[-1.3877787807814457e-16, -0.0, 1.0]	[1.0, 0.0, 0.0]
20	PLANE			T	[1.5959455978986622e-14, 25.0, -39.99999999999998]	[1.3877787807814454e-16, 0.0, -1.0]	[-1.0, 0.0, -1.3877787807814454e-16]

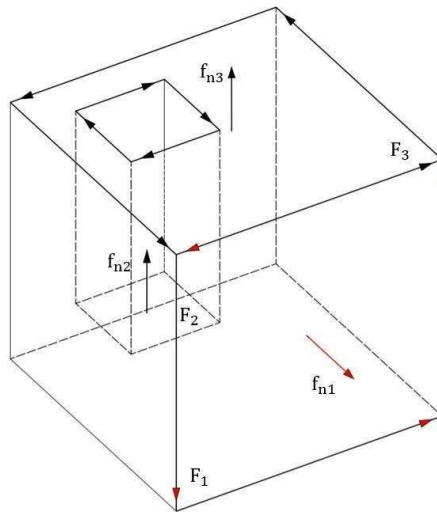
18. Ábra: Adatok a felhasználói felületen

IV.2 Alaksajátosság felismerés

IV.2.1 Felhasznált elméleti alapok

IV.2.1.1 Prizmatikus alaksajátosságok

Bhandarkar és Nagi [4.] tüzetesebben foglalkozott a problémával és részletezték a megoldásukat. Hogy képesek legyünk alaksajátosságokat felismerni egy STEP fileban meg kell találnunk a korábban említett kulcs entitásokat és felépíteni a fa struktúrát. Elsőnek meg kell találnunk egy zárt burkot majd meg kell vizsgálnunk ennek oldalait. Mindhez tartozik egy normális, valamint egy külső határolóvonal és opcionálisan egy vagy több belső határolóvonal, amelyek megléte belső formai alaksajátosságot jelent. Ha megvizsgáljuk ezeket a határvonalakat akkor zárt élláncokat találunk, amelyek orientált élekre bonthatók. Minden élt a kezdő- és végpontjának sorrendjével meghatározott irányultság jellemez és EDGE_CURVE paramétere megmondja a típusát.

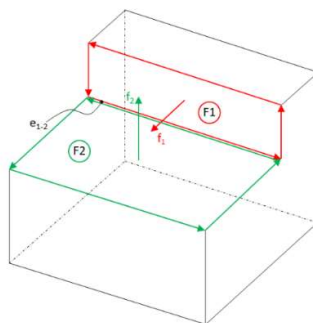


19. Ábra: Orientációs szabályszerűségek

A már definiált fa legtöbb szintje rendelkezik operátorokkal, hogy irányt adjon a geometriának. A 19. Ábrán látható az egyik legfontosabb szabálya a B reprezentációnak. Az oldalhatároló élek mindig úgy vannak orientálva, hogy az anyag a bal oldalukon legyen. Így tehát azok az élek, amelyek külső határolásban vesznek részt az óramutató járásával ellentétesen haladnak, míg a belső határolóvonalak az óra mutatójával megegyezően. Ennek a struktúrának köszönhetően a furatok és zsebek megtalálhatóak és ahhoz, hogy eldöntsük, hogy ezek átmenők-e vagy nem meg kell néznünk a kapcsolatukat más oldalakkal is. Hogy képesek legyünk megtalálni az olyan külső formai alaksajátosságokat, mint például a lépcső vagy a zseb, a konvexitást és konkávitást

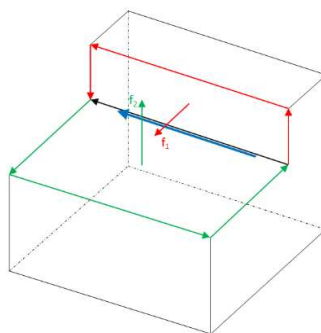
kell segítségül hívni. Ezek a felvett normálisok kapcsolatából állnak elő. Attól függően, hogy mennyire komplex egy alaksajátosság a számítás összetettebb lehet.

A lentebb részletezett Boolean alapú topológia segítségével meg tudjuk állapítani a felülethez tartozó élek konvexitását és konkávitását, így képesek vagyunk olyan alaksajátosságokat megtalálni, mint például zsebek, hornyok, lelapolások.



20. Ábra: Az elképzelt orientáció

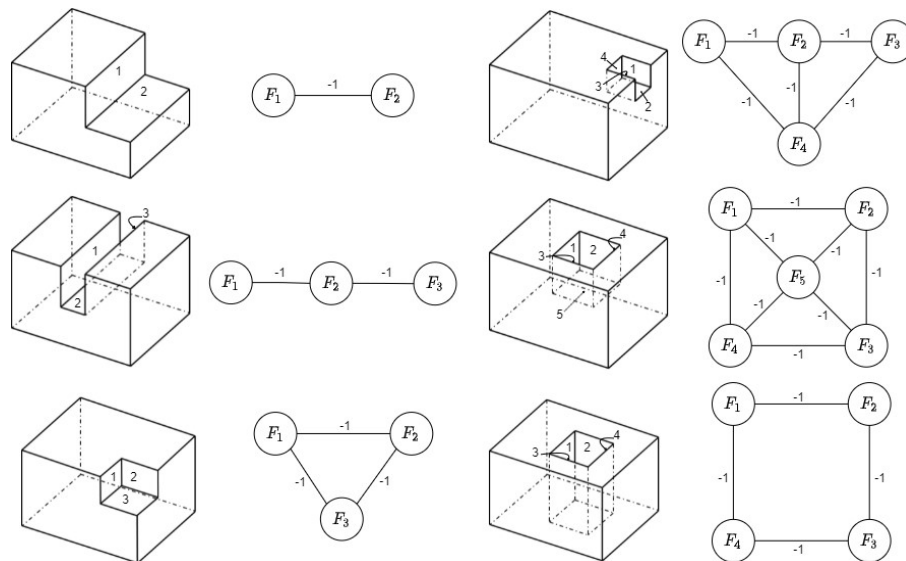
Ahogy a 20. Ábrán is látható minden felület úgy van definiálva, hogy a normál vektora kifelé mutasson az anyagból, így az élek jobb sodrását előállítva, belátható viszont, hogy így egy darab élnek két (általá összekötött) különböző felülethez viszonyítva, ellentétes irányba kell állnia. Egy él irányultsága a kezdő és végpontjából adódik, ez alapján az alsó ábrán látható, hogy a két jelölt felületet összekötő él (fekete), irányultsága a tőle balra eső felülethez igazodik (zöld). Ahhoz, hogy a tőle jobbra eső felülettel is előálljon a jobbsodrású konvenció, szükségesek az irányított élek Boolean attribútumai, amelyek az irány negálására szolgálnak, így egy záródó élhurkot előállítva.



21. Ábra: A valós orientáció

Ahhoz, hogy két felület konkáv/ konvex kapcsolatát determinálhassuk, a köztük feszülő élt és a normálvektorokat kell vizsgálnunk. Az $f_1 \times f_2$ keresztszorzatból előáll egy a közös éllel párhuzamos egyenes, amennyiben irányultságuk megegyezik, akkor az két felület konkáv szöget zár be, amennyiben ellentétes akkor konvexet. Fontos, hogy a vektorok keresztszorzata nem kommutatív, a konvenció mindig úgy néz ki, hogy az él eredeti irányultságától jobbra eső felület normálvektora áll az első, még a balra lévő a második helyen. A 21. Ábrán látható, a keresztszorzás eredményeként kapott kék vektor, melynek iránya megegyezik a felületeket összekötő fekete él irányával, helyesen konkáv szöget definiál a két felület között.

Az általunk használt algoritmus is szemléltethető gráf struktúrában. Első lépésben ábrázoljuk külön-külön az összes felületet, amely konkáv kapcsolatban áll egy másikkal, mint csomópont, ezt követően hozzájuk rendeljük a hozzájuk tartozó, konkáv kapcsolatot biztosító éleket. Majd az egyesítésük következik, amennyiben két csomópont is rendelkezik egy azon konkáv összeköttetést biztosító éllel akkor egyesíthetőek gráf formájában, azaz egy alaksajátosságot alkotnak. Ezen az elven elindulva az összes csomópontot páronként vizsgálva véges iterációban eljuthatunk az egymástól teljesen elkülönülő gráfokig, amelyek ekvivalensek egy alaksajátossággal. A 22. Ábrán láthatóak a korábban tárgyalt alaksajátosság típusok általunk használt gráfos felépítése. A -1-ek a konkáv kapcsolatot jelentik.



22. Ábra: Az feature-ök által definiált gráf

Az elkészült gráfokat további vizsgálatokkal, előre definiált minták segítségével párosíthatjuk adott alaksajátosságokhoz.

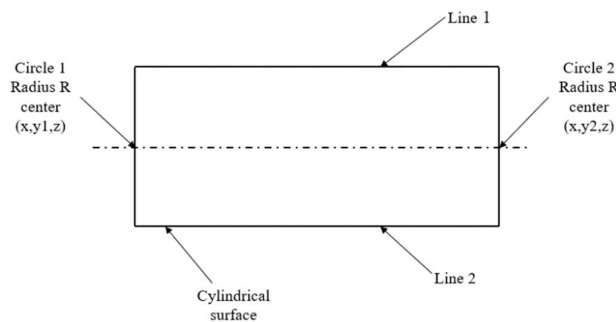
IV.2.1.2 Hengeres alaksajátosságok

Az eddig tárgyalt és általunk is használt gráfalapú módszereket alapvetően prizmatikus alaksajátosságok megtalálására fejlesztették ki. A teljesen hengeres alaksajátosságok felismeréséhez egy kevésbé komplex algoritmust használtunk. Itt nem az volt a kiindulás, hogy az adott alaksajátosság, hogyan módosítja az adott geometriát, hanem az a megállapítás, hogy amennyiben teljesen hengeres felületet találunk, az gyártástechnológiai szempontból nem lehet más, mint hengeres alaksajátosság, azaz csap vagy furat. Ennek módszere a gráf alapú megközelítéstől és a konvex/ konkáv vizsgálatától elkülönül, ezért azoktól teljesen elválasztandó. Tekintve, hogy furat esetén a megmunkálás módja alapvetően fűrés, amely ciklusidejének meghatározásához a furatok száma, átmérője, mélysége és kiosztása a mérvadó, elegendő a vizsgálatban a hengeres felületek geometriai leírására összpontosítani.

A furatok, valamint a csapok is azonosíthatóak a cilindrikus felületekhez tartozó algoritmus használatával. Az irodalomban kevés szó esik az olyan alaksajátosságok felismeréséről, amelyek részben vagy teljesen hengerek, utóbbi esetben a Mellashwari [6.] által megállapított törvényszerűségeket vettük alapul.

Egy hengeres felület felismerésének szükséges feltételei, lásd 23. Ábra:

- szabály 1: EDGE_CURVE entitásnak tartalmaznia kell egy circle, line, circle, line vagy line, circle, line, circle élsorozatot
- szabály 2: az EDGE_CURVE körök rádiusza meg kell egyezzen a cylindrical atribútum rádiuszával
- szabály 3: A körlapok egytengelyűek, innen következik, hogy távolságukból számítható a hengeres felület magassága
- szabály 4: amennyiben a három szabály teljesül a felület mindenképpen cilindrikus



23. Ábra: A cilindrikus szabályszerűség

A felületi normálisok felhasználásával el bírjuk dönteni, hogy furatról vagy csapról beszélünk, amennyiben az előbbiről, meg kell vizsgálnunk a többi felületet, hogy el bírjuk dönteni, hogy ez egy átmenő vagy zsák furat.

IV.2.2 Megvalósítás

IV.2.2.1 Prizmatikus alaksajátosságok

A programunk felépítése szorosan követi az elméleti részben elmondottakat. Az általunk használt algoritmus merít a hibrid feldolgozási megfontolásokból, kiindulásként a gráf alapú módszereket használtuk, viszont más megközelítéssel. Mi nem szomszédsági mátrixot használunk gráf reprezentálására, hanem illeszkedési mátrixot, ennek előnye, hogy a sor-oszlop összerendelésből egyértelműen meghatározható a kapcsolat mellett a kapcsolatban résztvevő él is. Az adatkinyerés során minden felülethez kinyertük a hozzájuk tartozó éleket, azonban tekintve, hogy egy él két felülethez is tartozik, az adatbázisunkban minden él kétszeres multiplicitással szerepel. Először létrehozunk egy egyedi él listát, itt már egy él csak egyszer szerepel. A listát összehasonlítva a STEP adatbázissal pontosan meghatározható, hogy az adott él mely két felületet kapcsolja össze. Kihasználva azt a törvényszerűséget, hogy az él a tőle balra eső felület sodrásához igazodik, a Boolean értékek vizsgálatával már ebben a lépésben információt kapunk a felületnormálisok keresztszorzásának sorrendjéről. Ezt az összetartozást felvesszük az illeszkedési mátrixba úgy, hogy egyértelműen jelöljük, hogy az egyes kapcsolatoknál melyik felület normálvektora lesz a keresztszorzásban az első és melyik a második helyen. A következő lépésben elvégezzük a szükséges matematikai műveleteket figyelve arra, hogy a STEP által leírt normális irányultsága szintén egy Boolean változó függvényében lehet ellentétes. Az adott felületek közötti konkáv/ konvex kapcsolat megállapításra, majd tárolásra kerül az illeszkedési mátrixban. Az illeszkedési mátrixban 1-esel

jelöltük a konvex és -1-essel a konkáv kapcsolatokat. Ennek segítségével egyetlen mátrix elemzésével olyan információkhoz juthatunk, hogy egy adott felület mely más felületekkel kapcsolódik, melyik éllel, és hogy konvex vagy konkáv szöget zárnak e be. Az alapvetésünk ugye az volt, hogy ahol konkáv kapcsolat fellelhető ott biztosan valamilyen alaksajátosság található. Ezt követve meghatároztunk minden felületet és élt, ami potenciálisan részt vesz egy alaksajátosság kialakításában. A keresés itt kizárólag a konkáv szöget bezáró felületekre és az őket kapcsoló élekre korlátozódik, így ezeket csak “ömlesztve” kapjuk meg a programból. A csoportosítás algoritmusa (24. Ábra) már nem triviális, hiszen az illeszkedési mátrix többszöri kereszthivatkozással kapjuk meg az összetartozó éleket és felületeket.

```
while len(F)!=0:
    VIZSGALT_FACE=[]
    VIZSGALT_EDGE=[]
    VIZSGALT_FACE.append(F[0])
    while True:
        cond1=VIZSGALT_FACE.copy()
        cond2=VIZSGALT_EDGE.copy()

        for i in VIZSGALT_FACE:
            for j in np.where(F==i)[0]:
                if E[j] not in VIZSGALT_EDGE:
                    VIZSGALT_EDGE.append(E[j])

        for i in VIZSGALT_EDGE:
            for j in np.where(E==i)[0]:
                if F[j] not in VIZSGALT_FACE:
                    VIZSGALT_FACE.append(F[j])

        if cond1==VIZSGALT_FACE and cond2==VIZSGALT_EDGE:
            break

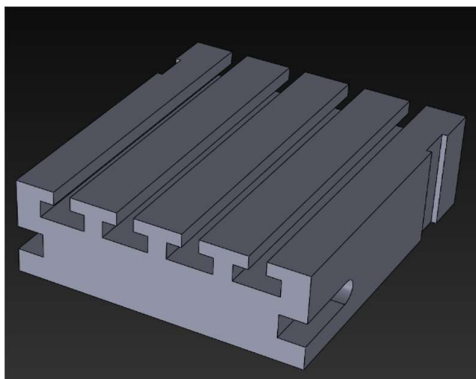
    FEATURE_F.append(VIZSGALT_FACE)
    FEATURE_E.append(VIZSGALT_EDGE)

    for i in VIZSGALT_FACE:
        while i in F:
            F.remove(i)

    for i in VIZSGALT_EDGE:
        while i in E:
            E.remove(i)
```

24. Ábra: A csoportosítás algoritmusa

A csoportot alkotó felületeket és éleket pároként vizsgálva szétválaszthatóak az elkülönülő alaksajátosságok. Majd a felületek kapcsolatából, elhelyezkedéséből, bezárt szögéből, valamint az élek egymáshoz képesti viszonyából meghatároztuk az alaksajátosság típusát méreteit és a megmunkálás irányát.



Feature based STEP processing

Adatbázisok

STEP file: [C:\Users\jmedf\Documents\test\STEP]

Data extraction

Felületek	Külső határoló vonalak	Határoló vonalak	Purat táblázat	Prizmatikus feature táblázat	Esztergált felületek	Megmunkálási információk
Type	L[mm]	h1[mm]	h2[mm]	b1[mm]	b2[mm]	
T-horony	59.721050000000005	3.4934700000000002	2.7763899999999992	8.5305600000000001	3.16850000000000016	
T-horony	59.721050000000005	3.4934700000000002	2.7763899999999992	8.53056	3.1685	
T-horony	59.721050000000005	3.4934700000000002	2.7763899999999992	8.53056	3.1685	
T-horony	59.721050000000005	3.4934700000000002	2.7763899999999992	8.53056	3.1685	
Retszhorony	13.960639999999998	5.0		6.5162600000000001		
Átmenő horony	40.0	1.0		6.6181300000000001		
Retszhorony	13.960639999999998	5.0		6.5162600000000001		
Átmenő horony	40.0	1.0		6.6181300000000001		

25. Ábra: A megtalált alaksajátosságok

Itt fontos megjegyeznünk, hogy a geometriai megállapításokat megpróbáltuk minél általánosabb formában megfogalmazni, leginkább az entitások kapcsolatát felhasználva (25. Ábra). Konkrét értékeket, párhuzamosságot esetleg merőlegességet csak akkor írunk elő, ha az az adott feature definíciójához szorosan kapcsolódik.

```
elif nof == 3:
    if (np.array(knkveof)==np.array([1,1,2])).all() and (FACE_EDGE[FEATURE_F[feature]][:,FEATURE_E[feature]]!=1).all() \
    and (FACE_EDGE[FEATURE_F[feature]]!=2).all():
        prismatic_feature["Type"].append("Átmenő horony")
        primerface = 0
        for e in range(len(feature_edges)):
            L = 0
            if edge_length(feature_edges[e]) > L:
                L = edge_length(feature_edges[e])
            if e == len(feature_edges) - 1:
                prismatic_feature["L[mm]"].append(L)
        for i in range(len(FACE_EDGE[feature_faces])):
            if len(np.where(FACE_EDGE[feature_faces][i] == -1)[0]) == 2:
                primerface = feature_faces[i]
                prismatic_feature["b1[mm]"].append(
                    np.array([edge_length(b) for b in np.where(FACE_EDGE[feature_faces][i] == 1)[0]]).max())

        for i in np.array(EDGE_LIST)[np.where(FACE_EDGE[np.where(
            FACE_EDGE[:, feature_edges[0]] != 0)[0][np.where(
                FACE_EDGE[:, feature_edges[0]] != 0)[0] != primerface] == 1)[1]):
            for j in i:
                if (j == EDGE_LIST[feature_edges[0]][0]).all():
                    prismatic_feature["h1[mm]"].append(np.linalg.norm(i[1]-i[0]))

        prismatic_feature["h2[mm]"].append("")
        prismatic_feature["b2[mm]"].append("")
```

26. Ábra: Featurók csoportosításának kódrészlete

Az kód elején látható a nof (number of faces) változó mely az alaksajátosságokat alkotó felületek számát adja vissza. Amennyiben ez 3, belépünk a 3 felületű alaksajátosságok alcsoportjába. A kapcsolati vektor, ami jelen esetben (1,1,2) pedig a felületekből kiinduló konkáv élek számát jelöli. Látható, hogy ez az eset 22. Ábra bal oszlopának középső esetét reprezentálja, tehát megállapítható, hogy ez egy átmenő horony.

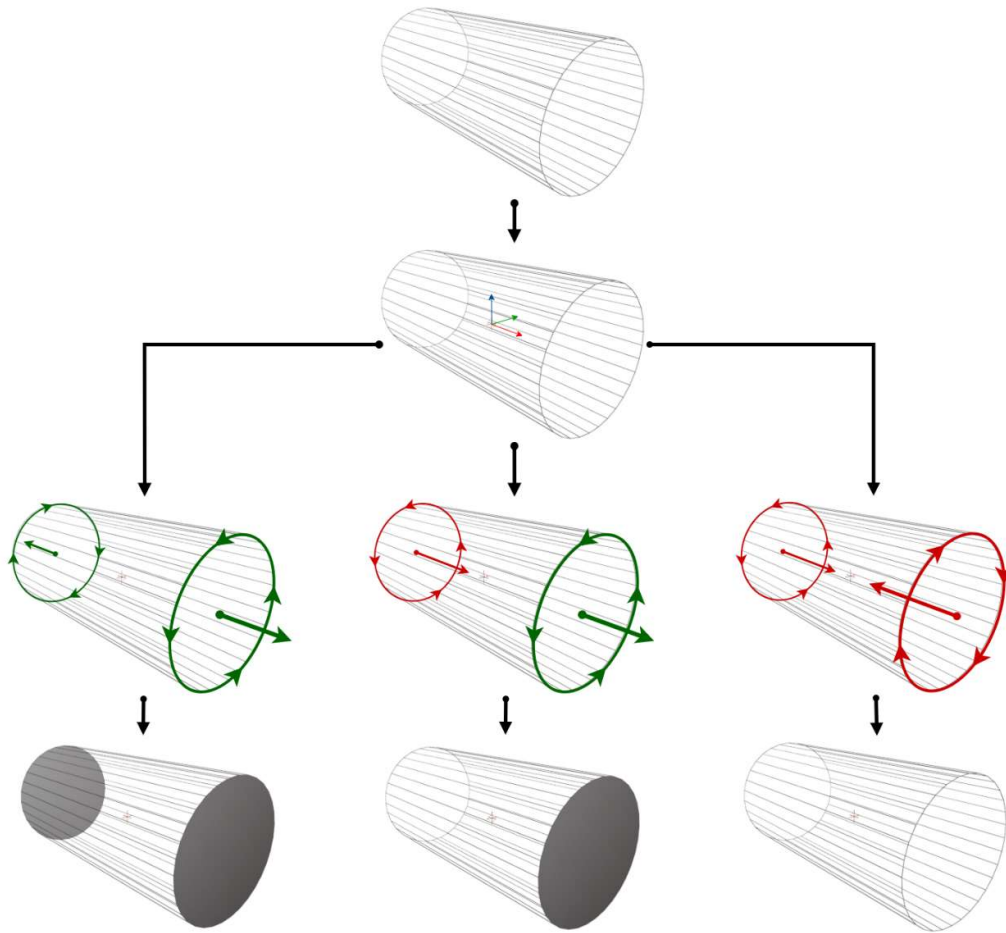
IV.2.2.2 Hengeres alaksajátosságok

A hengeres felületek a STEP fájlban az ADVANCED_FACE attribútumainak vizsgálatával könnyen kiszűrhetők. Amennyiben az őket felépítő élek EDGE_CURVE paraméterei circle, line, circle, line/ line, circle, line, circle struktúrát követnek, levonhatjuk a konklúziót, hogy hengeres alaksajátosságot találtunk. Ezeknél az alaksajátosságoknál a nehézséget az okozza, hogy eldöntsük róluk, hogy a hozzájuk tartozó térfogat negatív vagy pozitív-e, azaz csapról vagy furatról beszélünk, utóbbi esetében pedig az is determinálandó, hogy átmenő vagy zsák.

A szükséges algoritmus tehát a teljesen hengeres felületek megkeresésével kezdődik, itt először a felülettábla rekordjait végig pásztázvamegkeressük a Cylindrical felülettípusokat. Amennyiben az adott felületet meghatározó élek kielégítik a fent említett szabályszerűségeket, akkor megbizonyosodtunk róla, hogy a vizsgált entitás hengeres alaksajátosság. A megtalált elemet tovább vizsgálva már csak azt kellett megállapítanunk, hogy melyik típusról van szó. Itt azt a szabályszerűséget vettük alapul, hogy a felületek határológörbéi mindig olyan irányultságúak, hogy tőlük az anyag balra essen. Ebben a megközelítésben a nehézséget az okozta, hogy a kör felületek határoló élei kizárólag két pont megadásával állnak elő, melyek teljes kör esetén rend szerint egybeesnek. Ebből kifolyólag a határoló görbét a megszokott módon nem tudtuk vizsgálni, ezért a normálvektorokhoz képesti jobbsodrású konvenciót kihasználva tudtuk meghatározni, hogy a henger mely vége/ végei zártak. Hagyományos, azaz leválasztó megmunkálást feltételezve ez a megközelítés minden esetben helytálló, mivel egy átmenő furat mindig úgy áll elő, hogy a henger két vége nyitott, míg zsákfurat esetén csak az egyik. Ezzel szemben csapok esetén, a felület mindkét vége zárt. Az additív technológiájú gyártás szemszögéből nézve azonban, elképzelhető olyan bennszülött, azaz void feature (9. Ábra), ahol a szobán forgó törvényszerűség nem helytálló. A dolgozatban az additív technológiák megközelítésére nem tér ki, ezért a tárgyalt algoritmus implementálható. Az általunk elkészített programban ez úgy jelenik meg, hogy a két körlap közé egy hengeren belüli pontot definiál, ezzel egyértelműsítve, a hengerbe befelé és belőle kifelé irányok meghatározását. Ezután a dolgunk már csak annyi, hogy a körlapok normálvektorait összehasonlítsuk ezen irányokkal, így determinálva a hengeres alaksajátosság típusát.

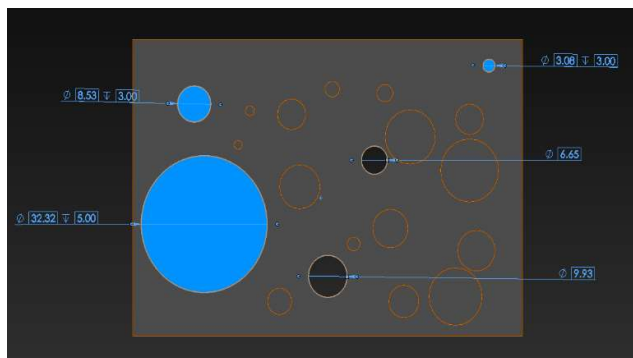
A folyamat az alábbi 27. Ábrán, fa struktúrában követhető nyomon. A hengeres felület megtalálásától indulunk, majd a benne elhelyezett ponthoz köthető koordinátarendszer és a körlapok normálvektoraianak irányának viszonyától függően az ábra három irányba ágazik. Ezek az irányok pedig a három lehetséges alaksajátossággal lesznek ekvivalensek. Amennyiben a körlapok normálvektorai kifelé mutatnak a belső ponthoz képest, akkor csapot találtunk, az

ellentétes scénárióban, azaz amikor befelé mutatnak átmenő furatról van szó, míg abban az esetben, ha az egyik kifelé a másik pedig befelé mutat, akkor zsákfuratról.



27. Ábra: A hengeres felületek felismerésére implementált logika

Az általunk használt logikát, több munkadarabon is leteszteltük, az algoritmusunk képes megtalálni a hengeres alaksajátosságokat és kinyerni az őket jellemző szükséges geometriai információkat. A 28. Ábrán látható egy példa, ahol a feldolgozott és rendszerezett adatmennyiség egy adatbázisban, úgynevezett furattáblázatban kerül a felhasználói felületre, így lehetővé téve a felhasználóknak az összes fontos geometriai információ áttekinthetőségét.



Feature based STEP processing

STEP file: C:\Users\medff\Documents\test.stp

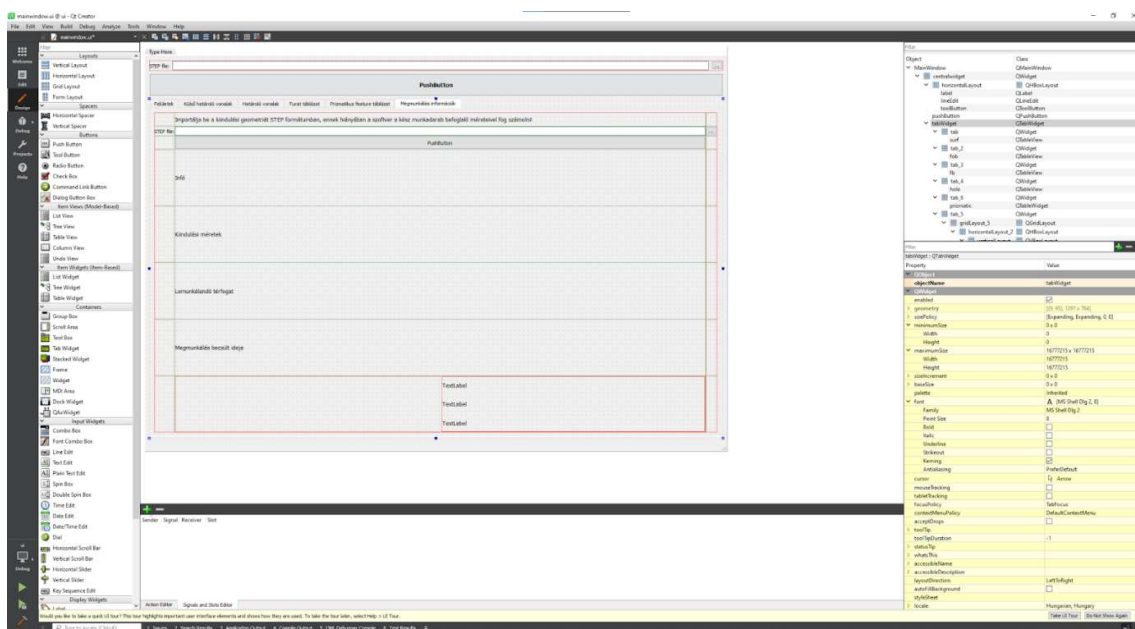
Data extraction

Surfaces	Face outer bounds	Face bounds	Hole table	Manufacturing information
Type	Hole length	Hole diameter		
Blind hole	9.0	4.0914034067231		
Blind hole	9.0	14.86232442172852		
Blind hole	1.0	7.6740426663702		
Blind hole	1.0	7.24262703206116		
Blind hole	1.0	13.62945707805082		
Through hole	10.0	6.65099938810302		
Through hole	10.0	9.9254673253778		
Blind hole	5.0	12.84281338106948		
Blind hole	5.0	3.63993220569888		
Blind hole	5.0	2.30209534823442		
Blind hole	5.0	2.1058002516069		
Blind hole	5.0	3.23064250937438		
Blind hole	5.0	9.59084754806		
Blind hole	5.0	32.318140570374		
Blind hole	3.0	9.05938476993044		
Blind hole	3.0	7.162238884366		
Blind hole	3.0	3.07957616510186		
Blind hole	3.0	6.17928973225262		
Blind hole	3.0	8.52847517027638		
Blind hole	3.0	10.360733659432		

28. Ábra: Furatok felismerése

IV.2.3 Program bemutatása

Mint már említettük, a program Python környezetben íródott, hogy a különböző gépi tanulással segített predikciós eljárás később egyszerűen implementálható legyen. A Pythonon belül a re (Regular Expression), numpy, pandas és ui könyvtárakat használtuk. A kezelőfelületet Qt Creator segítségével hoztuk létre (lásd 29. Ábra), amit a Python környezetben belül PyQt5 kezelőkönyvtár segítségével fordítottuk.



29. Ábra: User interface készítése

Jelen dolgozat leghangsúlyosabb részei az algoritmusok kialakítása, fejlesztése, finomítása, az irodalomból vett minél több ötlet kipróbálása és átültetése a gyakorlatba. Ennek fényében a kezelőfelület fejlesztésére kevesebb időt szántunk, leginkább annak bemutatására, hogy milyen információkat és hogyan szeretnénk közölni a felhasználóval. Első közelítésben a magunk ellenőrzésére még megjelenítettük a STEP adatbázist és az alaksajátosságokat tartalmazó táblázatokat, de ezeket a jövőben csak a lényegi információk kiszámolására szeretnénk csak használni. A kezelőfelület tehát igen letisztult, ahogy a 30. Ábrán is látható

30. Ábra: Kezelőfelület

A programunk jelenlegi kezelőfelületén minden információ különböző füléken van tárolva. A “Felületek” fül a STEP adatstuktúra fő táblázata, ez gyűjti és sorolja a különböző felületeket (ADVANCED_FACE). A “Külső határoló vonalak” azaz FACE_OUTER_BOUND-ok a külső míg “Határoló vonalak” azaz FACE_BOUND-ok a belső határoló vonalait veszi sorra a felületeknek. A “Furat táblázat” és a “Prizmatikus feature táblázat” pedig a felismert alaksajátosságokat szedi táblázatos formába. A fő felületünk a “Megmunkálási információk” nevet viseli. A felhasználók számára későbbiekben is elérhető információkat tartalmazza, mint a becsült idő, becsült költség különböző paraméterváltoztatások hatása, esetleg eszköz rendelkezésreállással kapcsolatos és egyéb információk. A felületen megjelenik egy újabb importálási lehetőség, ahová a kiindulási geometriát tudjuk behívni. A “Start prediction” gomb megnyomásával pedig elindul az idő- és költségbecslés.

V Gyártási időbecslés alapjai

V.1 Irodalomkutatás a feature alapú gyártási időbecslésről

V.1.1 Becslés megadásának lehetőségei

Ahogy a dolgozatunk bevezetésében is említettük a megmunkálási idő több mérnöki feladat alapinformációja. Ezen értékeket az iparban a gyártástechnológus szakember kalkulálja, aki különböző segítséget kaphat olyan automatizált algoritmusoktól, amelyek a feladatai bizonyos részét elvégzik. A gyártási időt és költséget a munkadarabra vonatkozó információkon túl több, együttesen fennálló körülmény befolyásolja. Gyakori ilyen problémák közé tartozik [21.] [22.]

- Aktuális alapanyagárak
- Információhiány a gyártandó termékről
- Megrendelő szűkös határideje
- Szerszámgép állapota
- Kommunikációs nehézségek a felek között

Belátható, hogy a gyártási idő függ a gyártási körülményektől, így figyelembevételének mértéke befolyásolja az időbecslés pontosságát. Minél pontosabb eredményt szeretnénk, annál pontosabb, részletesebb, ebből következően drágább módszert kell alkalmaznunk. Nem érdemes tehát a feladat megkövetelte pontosságot meghaladó erőforrást felhasználni. A minél kisebb ráfordítással elérni kívánt megfelelő minőségű eredmény elérésének megoldása a becslés automatizálása. Ahhoz, hogy egy műveletet automatizálni tudjunk, ismernünk kell a becsléshez szükséges bemenő adatokat, majd ezeket törvényszerűségekkel párosítva kiszámíthatóvá válik a termék előállítás ideje.

Több megközelítésből lehet dolgozni, amelyek ugyan eltérőek, de nagyságrendileg jól korrelálnak egymáshoz. Tekintsük át a leggyakoribb jelenlegi metódusokat.

NC program alapján történő becslés

Az NC programos módszer a szerszámpályák hosszának, valamint az előtolási sebességeknek a figyelembevételével számítja ki a megmunkálási időt. Esetenként eltekint a gép további jellemzőitől, így a CAM rendszer nem tud pontos becslést elérni. Mindazonáltal a geometriai feltételek, amely a megmunkálási folyamat információihoz kapcsolódnak szintén fontos hatástényezők az NC megmunkálási idő becslésének. Különösen hasznos a valós idejű gyártási folyamattervező és ütemező rendszerekben. A közelítés könnyen megvalósítható, így ezt a

módszer széles körben alkalmazzák a kereskedelemben elérhető szimulációs szoftverekben, mint például a Master CAM.

NC program alapú megközelítés gépi teljesítmény figyelembevételével

Az NC módszer az előző pontban tárgyaltak szerint érvényes ezen eljárásban is. Gépi paraméterek szerint különféle módszerek állnak rendelkezésre ebben a kategóriában. A két leggyakrabban használt típusnál az első- gépparaméterek és gépkonfigurációs fájlok ismeretében végzi a becslést, míg a másik a megmunkálási munkakörülmények szimulációját használja fel a becsléshez.

Hasonlóság alapú becslés

A hasonlóságot az alkatrész teljes egészében is lehet vizsgálni. Ilyenkor a módszer a késztermék minden paraméterét figyelembe veszi és ennek függvényében keresi meg a rá legjobban hasonlító adatbázisban tárolt alkatrészt. Ez a fajta becslési megoldás képes egy alaksajátosságot is felismerni a gyártandó alkatrészen és ezt csak a vele azonos típusú feature-ökkel fogja hasonlítani, azaz például egy zseb csak zsebekkel kerül összehasonlításra. Az adatbázis tartalmaz minden gyártandó alaksajátosságra több különböző mintát és a mintákhoz rendelt forgácsolási időt, így a példánknál maradván az adatbázisban eltárolt minden zsebknek ismert az előállítási ideje. A módszer a felismert alaksajátosság paramétereit veti össze az eltárolt feature-ök paramétereivel és keresi a legjobb egyezőséget. Miután megtalálta a közel azonos párt, a gyártandó alaksajátosságot párosítja az ismert feature megmunkálási idejével. A rendszer stabil és megbízható működése az adatbázis méretétől függ, mert a becslés annál jobb lesz minél több adat áll rendelkezésre. Mesterséges intelligencia hasonlóságelemzéssel karöltve fel tudja ismerni az alaksajátosságot, de hátránya a roppant nagy adatbázis igény, továbbá feltanítása hosszadalmas, alkalmazása függ a környezettől, azaz új helyre, vagy megmunkológépre kerülve, újra fel kellene telepíteni.

Anyagleválasztáson alapuló becslés

Yang és Lin a megmunkálási időt elsősorban az eltávolított anyag térfogata alapján közelítette. Ehhez megmunkálási jellemzőket vettek figyelembe, mint előállítási idő, gyártási jellemzők, megmunkálandó anyag és szerszámok. [21.]

V.1.2 Az általunk használt megfontolás

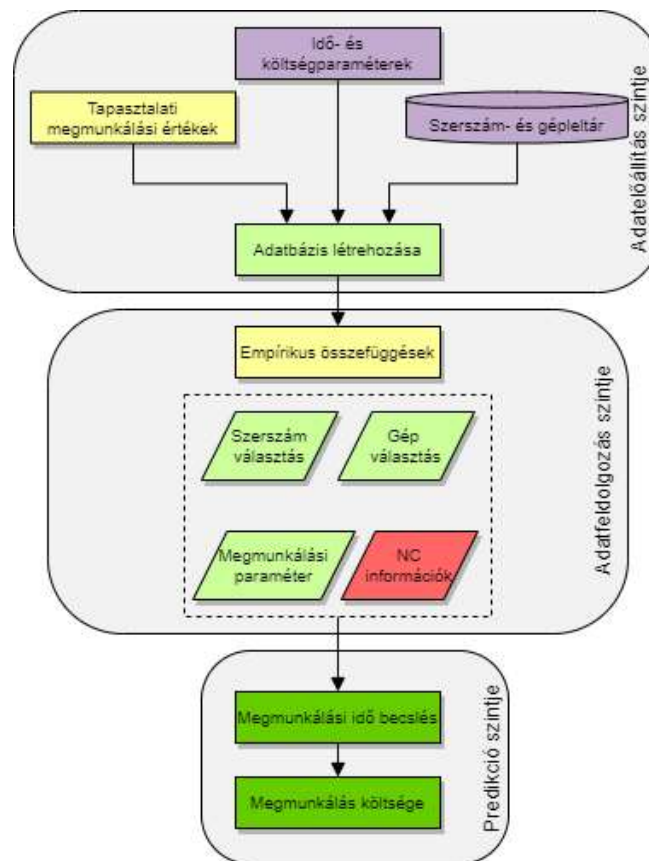
Az említett becslési eljárások több alapra épülnek, mégis közel hasonló időértéket képesek válaszolni. A gyártófüggő szoftverek ki tudják számítani a megmunkálási időt a szerszámpályák és mellékmozgások figyelembevételével, amivel pontosabb becslést adnak. Azonban figyelmen kívül hagyják a gép jellemzőit, amik szintén hozzájárulnak a pontos eredmények kalkulálásához. A legnagyobb negatívum az, hogy szükség van az NC programra, vagy legalább a szerszámpályákra, amik előállításuk pedig hosszadalmas és szakértelmet igénylő feladat. A folyamat gyorsaságát növelve és az egyszerűbb felhasználás irányába haladva szeretnénk a CAM rendszerek használatát kivenni a folyamatból. [24.][22.]

A korábban már tárgyalt meglévő becslési módszerek nem tudják teljes körben kielégíteni az időszorúság, a pontosság és a hatékonyság követelményét, mert az eljárási folyamat során elhanyagolnak egy-egy a technológiára jellemző kevésbé befolyásoló tényezőt. Mégis a hiányosság tudatában is képesek kellően alapos becslést adni a termék gyártását illetően. Az NC alapú megközelítés erőforrás igényes és szakértelmet kívánó feladat. A becsléshez szükséges egyik input adat a szerszámpályák hossza, amik egy felület megmunkálásakor több módon tervezhetőek, mint például horony esetén spirálszerű, trochoidális, vagy egyenes vonalú pályával. A pálya ilyen szempontból bizonytalanságot hordoz magában, mivel egyetlen felület előállítása is eljárásfüggő, azaz nem kapunk konkrét időértéket annak megmunkálására. Az anyagleválasztás bázisú megoldás nem szerszámpályákra támaszkodik, hanem a gyártási jellemzőkre, az anyag típusára és a szerszámmal végezhető forgácsolási paraméterekre épít. A programunk által kinyert technológiai és geometriai információk megadják a megmunkálás típusát és a feature paramétereit, így megalapozva a döntésünket, hogy az eltávolítandó anyagmennyiség megközelítési módszerét válasszuk megoldásunk alapjának. A mi módszerünk az egyedi, vagy a kis sorozatgyártásra fókuszál, ahol a szerszámok által megtett távolság kismértékű növekedése vagy csökkenése a teljes gyártás tekintetében nem jár számottevő idővel és költséggel. Nagyobb sorozatgyártásnál akár minden szerszám által megtett extra milliméternek nagy költségvonzata lehet, mivel ezek az extra utak összeadódva jelentős időnövekedést eredményeznek, ami többlet költséggel jár. [24.] [22.]

V.2 Szükséges adatok rendszerbe foglalása

V.2.1 Út a becsléshez

Mint korábban említettük, a megmunkálási idő több tényezőtől függ, amiket nehéz integrálni, hogy megmunkálási időt kapjunk. A becsléshez, azaz a folyamat utolsó lépéséhez több szinten keresztül jutunk el. Ezt a folyamatot az alábbi, 31. Ábra reprezentálja.



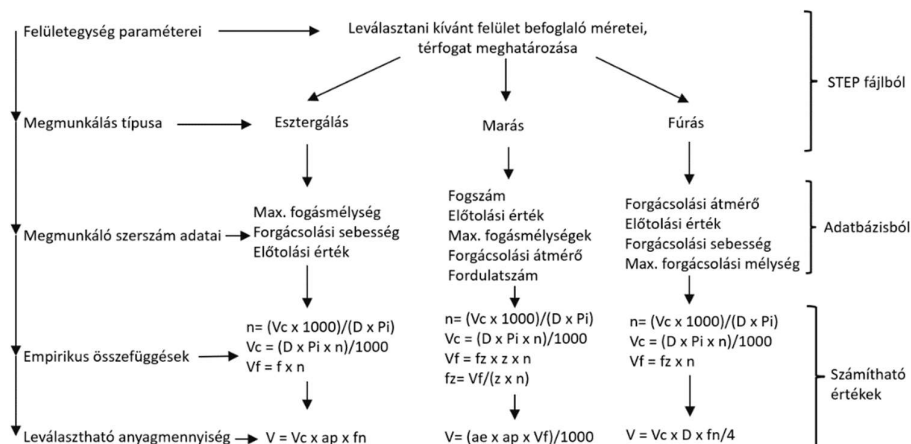
31. Ábra: A becslés folyamatábrája

A becslési folyamat három fő részből áll, melynek az első harmada az adatelőállítási szint. Ennek első lépésében egy szerszám és gépleltárat, valamint tapasztalati megmunkálási értékeket hívunk segítségül az adatbázis felállításához. A tapasztalati megmunkálási értékek, azaz a leggyakrabban használt gépi beállítások és szerszám paraméterek már egy előzetesen elkészített táblázatban vannak összefoglalva a szerszámgyártók által. Ezek mellett szükségünk van még az idő és költség megállapításához nélkülözhetetlen információkra, amit a STEP fájl nem tartalmaz és értékük gyártócégtől függ. A tapasztalati és megmunkálási értékeket, a szerszám és gépleltárat, valamint az idő- és költségparaméterek összehangolva létrehozuk az adatbázisunkat, mely három fő megmunkálásra összpontosít. A számunkra kulcsfontosságú

rész az adatfeldolgozási szinten történik, ugyanis itt empirikus összefüggéseket rendelünk a technológiákhoz. Ezen a szinten még négy további fontos lépést kell megtennünk a becslés megkezdéséig. A programunk segítségével detektáltuk a feature-ök geometriáját, ezek ismeretében megmunkáló eljárást illesztünk egy felület előállításához, azaz eldöntjük, hogy fúrásról, esztergálásról, vagy marásról van szó. A megmunkológép figyelembevételéhez szükségünk van a gép teljesítményére és a fajlagos forgácsolási erőre, amiből az anyagleválasztási térfogatot kapjuk a $Q = \frac{P}{K_c}$ összefüggéssel, ahol Q az időegység alatt leválasztható anyagmennyiség [cm³ /min], P a gép teljesítménye [W], K_c a fajlagos forgácsolási erő [N/mm²]. A megmunkáló eszközzel szembeni kívánalmak szoros összefüggésben állnak az előállítandó felület méreteivel. A szerszám a későbbiekben részletezett adatbázisban magával hordozza a rá jellemző felhasználási adatokat, amiből számolható, hogy mennyi a vele leválasztható anyagmennyiség [Q] adott idő alatt. A gépi és szerszám oldali Q értékét összehasonlítva megtudjuk, hogy mi lesz a szűk keresztmetszet. A kisebbik értékkel tovább számolva egy pontosabb becslési információt kaphatunk. A [31.Ábrán] piros színnel jelzett NC információk implementálása egy jövőbeni becslés pontosítási lehetőség. A predikció szintjén egyesítjük a technológiára jellemző összefüggéseket a szerszám paramétereivel egy felületre vonatkoztatva, amiből megkapjuk, hogy azt a bizonyos alak sajátosságát előállítani mennyi időbe telik. A folyamatot elvégezzük az összes felületre, majd a kapott időértékeket összesítjük a termék végleges előállítási idejének érdekében. Gyártási idő alapján elvégezhetjük a költségbecslést, amely csupán egy az idő ismeretében kalkulálható érték, így megtudjuk, hogy a STEP fájlban szereplő alkatrészt mennyi idő és mekkora költség előállítani. [21.]

V.2.2 A szükséges információk

Annak érdekében, hogy tisztábban átlássuk a gyártási folyamat során szükséges eljárásokat, technológiákat, eszközöket és a rendelkezésünkre álló információkat, ahol utóbbit a programunk által határozzuk meg, foglaljuk egy strukturált keretrendszerbe az eddigi ismereteinket. A teljes időbecsléshez szükséges információkat, így az alábbi 32. Ábra hivatott felvázolni.



32. Ábra: Az időbecsléshez szükséges információk keretbe foglalása

Ebből jól kivehető, hogy az elemző és feldolgozó programunk szolgáltat minden olyan adatot, amelyek technológiával párosítva bemeneti paraméterként jelennek meg a megmunkálási fázisnál. Minden egyes feature-nek ismerjük ezáltal az előállítási módját, azaz, hogy esztergáló, fúró, vagy maró eljárással készül. Amennyiben esztergálandó testről van szó, úgy a program biztosítja a számításokhoz a kezdeti megmunkálandó átmérőt, annak hosszát, valamint az elérni kívánt végleges átmérőt hosszszesztergálás esetén. Keresztesztergálásnál a kezdeti és végleges átmérőt, beszúrási tekintetében szintúgy ezen méreteket, valamint a beszúrási szélességét tudjuk meg. Marásnál meghatározza a szoftver a technológiát, hogy a marásnak milyen fajtáját kell elvégezni. Marhatunk sík felületet, ekkor ismernünk kell a munkadarab hosszúsági és szélességi paraméterét és az eltávolítandó anyagvastagságot. Vállak, hornyok zsebek megmunkálására külön marószerszám szükséges, ezeket rendszerint ujymaróval készítjük el. Váll kialakításához a programunk megadja az axiális és radiális mélységi értékeket. A hornyok vagy reteszek elkészítése abban az esetben egyszerű, ha rendelkezésünkre áll a horony vagy retesz szélességével megegyező átmérőjű marószerszám. Amikor ez nem áll fenn, akkor a kiválasztott megmunkáló eszközzel legalább még egy ismétlést kell tennünk a kívánt szélesség eléréséhez. Természetesen ezen alak sajátosságoknak is ki tudjuk olvasni mélységi értékét a feldolgozó segítségével. Zsebek előállításához is képes a szoftver megadni mindhárom jellemző méretet, de a zseb elhelyezkedése miatt így lehet szó zárt, vagy nyílt zsebről. E két típus megmunkálásban nem esik különbség. Fúrás esetén megtudjuk a programból a fúrási átmérőt, valamint annak mélységét. A megmunkálással elért átmérőt két úton érhetjük el. Az első esetben eleve a szerszám átmérője fogja adni a végleges méretet, ilyenkor telibe fúrásról beszélünk. Furatbővítő eljárás közé tartozik a felfúrás és a furatesztergálás. Utóbbinál a

fúrószerszámmal kimunkáljuk azt a méretet, ahol az esztergáló szerszám már képes a furatba hatolni és azt bővíteni.

V.2.2.1 Szerszámválasztás

Adatbázisra van szükségünk a szerszámok megmunkálási paramétereit illetően, hogy azokkal milyen körülmények között tudunk forgácsolni. Az adatbázis rendszerezi megmunkálási típus szerint a szerszámokat, így egyetlen helyről elérhetjük az adott feature-höz tartozó méretek ismeretében az előállításához szükséges eszközt. Megmunkáló eszköz kiválasztásánál már ismerjük a jellemző technológiát és befoglaló méreteket. Az összefüggések létrehozását megalapozta egy egyedi termékek gyártásával és forgácsolásával foglalkozó cégnél tett látogatásunk, ahol támogattak bennünket a program korrekt működéséhez elengedhetetlen információkkal és tapasztalati értékekkel. [23.]

Marás

A marási összefüggésünk a $Qt = (ap * ae * Vf)/1000$ képletből áll elő, ahol Qt a leválasztható anyagmennyiség [cm^3/min], ap az axiális fogásmélység [mm], ae a radiális fogásmélység [mm], Vf az előtolási sebesség [mm/min].

Amennyiben sík / homlokmarásra kerül sor, akkor a szerszámválasztáskor a leválasztandó felület kisebbik paraméterét vesszük figyelembe, azaz, ha le kell munkálni egy 100x60-as felületet, akkor a 60-as paraméter lesz a mérvadó. Az így kiválasztandó marószerszám átmérője legalább a kisebbik (jelen esetben 60mm-es) méret 125%-a. Ehhez az adatbázisban legközelebb eső eszköz a 80mm átmérőjű maró. Ez egy tapasztalaton nyugvó választás, ezért a százalékos érték módosítására szabad lehetőséget kínálunk az adatbázis módosítási lehetőségével. Akadnak olyan nagy munkadarabok, ahol ezen összefüggés nem áll fenn, ekkor a termelékenység érdekében a legnagyobb aktuálisan elérhető szerszámot használják. Ha több felületet kell homlokmarással előállítani, akkor a legkisebb marandó felületnek vesszük a bázisértékét és azzal iteráljuk ki a szerszámot. Nagy különbségek fennállásakor szerszámcserre eszközölhető, aminek a taktidejével szintén kalkulálni kell, melyre definiáltunk egy jól közelítő értéket az adatbázisban, ugyanakkor az eltérő szerszámcserélő megoldások miatt ezen értéket átírhatóvá tettük. Vállmaráskor a marószerszám átmérőjének a 20%-a (default érték, mely módosítható) a palástirányú fogásvétel (ae) és az axiális fogásmélység (ap) szerszámfüggő érték. Zsebek, szélesebb hornyok marását illetően, ahol utóbbiaknál legalább egy nyílt felülettel találkozunk, ott az alak sajátosságra jellemző marót választunk. Optimális esetben ez megegyezik a horony, vagy zseb szélességével. Amennyiben ez nem valósítható meg, úgy a

kisebbség befoglaló paraméter (kivéve Z irányban) 25%-a lesz a szerszámtármérőnk négyzetes, téglalap, vagy egyedi alakkal rendelkező felület esetében. Kör alapú zsebekhez pedig az átmérő 80%-a fogja biztosítani a választandó eszközt. Utóbbi számértékek megváltoztatására is biztosítjuk a lehetőséget. [23.]

Fúrás

A fúrási összefüggésünk a $Q_t = V_c * D * f_n / 4$ képletből áll elő, ahol Q_t a leválasztható anyagmennyiség [cm^3/min], V_c a forgácsolási sebesség, D a fúrási átmérő, n a fogankénti előtolás [mm].

Fúrás tekintetében szintén célszerű azonos átmérőjű fúrót választani az előállítandó furat átmérőjével. Adatbázisban eltárolt 20 milliméteres átmérőig alkalmazzuk, afelett felfúrással bővíthetjük a furatunkat, vagy belső furatesztergáló szerszámmal növelhetjük a belső méretet. Természetesen a telibefúrás értéke szerszám és megmunkológép függő, ezért szabadon átírható a default méret. [23.]

Esztergálás

Az esztergálási összefüggésünk a $Q_t = V_c * f_n * a_p$ képletből áll elő, ahol Q_t a leválasztható anyagmennyiség [cm^3/min], V_c a forgácsolási sebesség [m/min], f_n a fogankénti előtolás [mm], a_p az axiális fogásmélység [mm].

Forgásszimmetrikus alkatrészek terén esztergálással állítjuk elő az alaksajátosságot. Hossz és keresztesztergálást végezhetünk egy azon szerszámmal is. Be- és leszúrással elkészítéséhez a mélységét kell vizsgálnunk, ennek figyelembevételével választandó a megmunkáló eszköz. [23.]

V.3 Adatbázis bemutatása

Adatbázisunk létrehozásakor mérlegelnünk kellett az iparban alkalmazott megoldások gyakoriságát és részletességét, hogy egy adott eljárást milyen mennyiségben használnak az alkatrészek gyártása során. A részletesség alatt az eljárások arányosságát értjük, azaz a simítás egy felületarányos művelet, a nagyolás pedig térfogatarányos. Forgácsolás közben rendszerint nagyoló műveletelemeket alkalmazunk gyakrabban, mivel a pontosság csak az utolsó lépésben válik fontossá a termék méretét és pontosságát illetően. Három forgácsoló technológiát alkalmazunk a nyersanyagból az előgyártmány elkészítéséhez, melyekkel az összes általunk vizsgált alaksajátosság előállítható. Ezek a fúrás, esztergálás és marás folyamatai, ahol

nagylással távolítjuk el az anyagot. Ennek tudatában e három technológiára kerestünk szerszámokat és foglaltuk őket adatbázisba. A forgácsoló üzemben tett látogatásunkkor elmondták, hogy ők a milyen márkájú termékeket preferálnak és korábbi kutatásunk is arra engedett következtetni, hogy ők a szerszámgyártók terén közkedveltek. Természetesen több szerszámgyártótól vettünk katalógusadatokat, viszont az azonos eszközökre a megmunkálási paraméterek jól korreláltak egymáshoz. Példánkban a Sandvik termékein keresztül végezzük a predikciós eljárás fizikai részét. Az adatbázisunk ennek fényében a következő alakban épül fel:

Marás	Szerszám típusa Coromill 245 (45 fokos) marás	Átmérő D [mm]	Forgácsolási élek sz. z[db]	Fogankénti előtolás fn [mm]	Fordulatszám n [min-1]	Forgácsolási szélesség ae [mm]	Fogásmélység ap [mm]
Coromill 245 (45 fokos) marás	R245-032A32-12M	32	3	0,354	3520	25	6
Coromill 245 (45 fokos) marás	R245-040A32-12L	40	3	0,354	2870	31,25	6
Coromill 245 (45 fokos) marás	R245-050A32-12L	50	3	0,354	2340	38,46	6
Coromill 245 (45 fokos) marás	R245-063A32-12L	63	4	0,354	1850	50	6
Coromill 245 (45 fokos) marás	R245-080Q27-12L	80	8	0,354	1380	62,5	6
Coromill 245 (45 fokos) marás	R245-100Q32-12H	100	10	0,354	1170	71,43	6
Coromill 245 (45 fokos) marás	R245-125Q40-12H	125	12	0,354	858	100	6
Coromill 245 (45 fokos) marás	R245-160Q40-12H	160	16	0,354	641	125	6
Coromill 245 (45 fokos) marás	R245-080Q32-18H	80	5	0,283	1420	62,5	10
Coromill 245 (45 fokos) marás	R245-100Q32-18H	100	6	0,307	1360	35,71	10

33. Ábra: Síkmarás

Marás	Szerszám típusa Coromill Plura marás	Átmérő D [mm]	Forgácsolási élek sz. z[db]	Fogankénti előtolás fn [mm]	Fordulatszám n [min-1]	Forgácsolási szélesség ae [mm]	Fogásmélység egy fogásból [mm]	Max. Fogásmélység [mm]
Coromill Plura marás	1P230-0100-XA 1630	1	2	0,0038	58500	1	0,4	4,5
Coromill Plura marás	1P260-0150-XA 1620	1,5	3	0,0055	38400	1,5	0,125	6
Coromill Plura marás	1P260-0200-XA 1620	2	3	0,0087	28700	2	0,2	8
Coromill Plura marás	1P250-0250-XA 1630	2,5	2	0,0115	22200	2,5	1	12,5
Coromill Plura marás	1P250-0300-XA 1630	3	2	0,0155	18900	3	1,5	12,5
Coromill Plura marás	1P240-0350-XA 1630	3,5	4	0,021	17000	3,5	1,65	10,5
Coromill Plura marás	1P250-0400-XA 1630	4	2	0,025	14200	4	2,5	14,5
Coromill Plura marás	1P240-0450-XA 1630	4,5	4	0,029	13700	4,5	1,65	11,5
Coromill Plura marás	1P250-0500-XA 1630	5	2	0,0295	11600	5	3	16,5
Coromill Plura marás	1P240-0550-XA 1630	5,5	4	0,0335	11800	5,5	2,5	13,5

34. Ábra: Szármarás

Fúrás	Szerszám típusa CoroDrill460 fúrás	Átmérő D [mm]	Max. Fogásmélység [mm]	Fogankénti előtolás fn [mm]	Forgácsolási sebesség Vc [min-1]
CoroDrill460 fúrás	462.1-2000-112A0-XM	2	11,5	0,025	38,2
CoroDrill460 fúrás	462.1-3000-140A0-XM X2BM	3	14	0,1	38,6
CoroDrill460 fúrás	460.1-0400-030A1-XM GC34	4	32,5	0,115	113
CoroDrill460 fúrás	460.1-0500-038A1-XM GC34	5	40,7	0,115	113
CoroDrill460 fúrás	460.1-0600-045A1-XM GC34	6	48,9	0,153	111
CoroDrill460 fúrás	460.1-0800-060A1-XM GC34	8	65,2	0,2	111
CoroDrill460 fúrás	460.1-1000-075A1-XM GC34	10	81,5	0,25	111
CoroDrill460 fúrás	460.1-1200-090A1-XM GC34	12	97,8	0,28	107
CoroDrill460 fúrás	460.1-1400-105A1-XM GC34	14	114,1	0,28	107

35. Ábra: Fúrás

Esztorgálás (általános)	Szerszám típusa CoroTurn 107 (esztorgáló,	Forgácsolási sebesség Vc [min-1]	Fordulatonkénti előtolás fn [mm]	Fogásmélység ap [mm]		
CoroTurn 107	C5-SCLCR-35060-12	324	0,36	2,5		
CoroTurn 107	(CoroTmax-P)DCLNL 2020K 09	343	0,3	2		
Esztorgálás (belső felület)	Szerszám típusa CoroTurn 107 esztorgáló (belső felület)	Furat átmérő Df[mm]	Max. Mélység [mm]	Fordulatonkénti előtolás fn [mm]	Forgácsolási sebesség Vc [min-1]	Fogásmélység ap [mm]
belső felületi eszt.	A08H-SCLCL 06	10	32	0,2	359	1,5
belső felületi eszt.	A10K-SDUCL 07-ER	15	40	0,45	361	1,5
belső felületi eszt.	E16R-SCLCL 06-R	20	95	0,2	386	1,5
belső felületi eszt.	E16T-SCLCL 3	30,5	150	0,3	343	2

36. Ábra: Esztorgálás

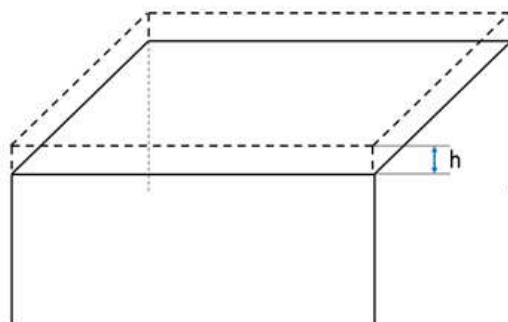
A három fő blokk, melyek egymás alatt sorokban helyezkednek el, a marás, a fúrás és az esztergálás. A táblázat következő elemei oszlopok formájában rendszereződnek. A címsorban szerepel a szerszám fajtája és típusa a nyomkövethetőség végett, a további értékek pedig az adott szerszám forgácsolási paramétereit hivatottak mutatni. Tulajdonképpen ezek az értékek a programunk működéséhez olyan további elengedhetetlen bemenő információk, melyek a megmunkálási technológia összefüggéseire illesztve biztosítják számunkra a leválasztandó térfogat [V] mérőszámát. A leválasztandó anyagmennyiség eléréséhez megmunkálási módtól függően szükségünk van a szerszám átmérőjére, maximális fogásmélységére axiális és radiális irányban, forgácsoló élek számára, előtolási sebességre és fordulatszámra. Az alaksajátosságok előállításához olyan szerszámokat kerestünk, melyekkel az ipari igényekben szereplő mechanikai elemek rendre legyárthatók, azaz hossz- és keresztesztergáló szerszámokat, fúrókat és furatbővítésre felfúrószerszámot, emellett furatesztergálót, valamint marás tekintetében homlokmarót, zsebek, vállak és hornyok elkészítéséhez szármarót és T-hornyokhoz alkalmazandó marószerszámot. [23.]

Adatbázisunk egy alapkészletet tartalmaz, amely alkalmas az említett feature-ök megvalósítására, de fenntartjuk a lehetőséget ennek bővítésére oly módon, hogy a felhasználó képes legyen rendszeren kívüli önnön választott szerszámot integrálni az adatbázisba a megmunkáló eszköz forgácsolási paramétereinek megadásával. Módosítási opciót kínálunk az általunk megadott default forgácsolási körülmények megváltoztatására, például egy fogásmélység csökkentésére. Az anyagjellemzők figyelembevételére arányszámokat alkalmazunk, melyek az C25-ös (175HB-s) acélt veszik referenciának és ehhez képest mutatjuk, hogy időegységenként mennyivel képes több anyagot leválasztani a szerszám azonos megmunkálási feltételek mellett. Mi a módszerünk bemutatáshoz az alumíniumra gyűjtöttük ki az arányokat forgácsolási tartományokra vonatkoztatva. Új anyag megmunkálására az acélhoz képesti forgácsolhatósági arányszámot szükséges megadni. Ennek egyszerűsége, hogy az új megadott arányszámmal súlyozásra kerül az időegység alatt eltávolítható anyagmennyiség.

VI Gyártási időbecslés és felhasználása a gyakorlatban

VI.1 Időbecslés különböző megmunkálásokra

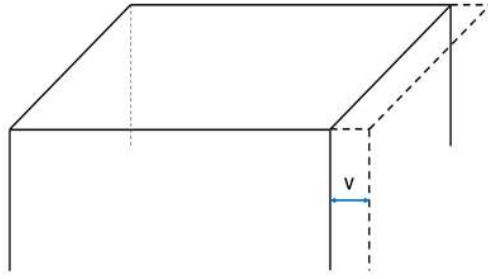
Sík felület homlokmarása



37. Ábra: Homlokmarás

Homlokmarásra szükségünk van az egyenletes és sík bázisfelület kialakítása végett. Az eljárás során előgyártmányból képezzük a majdani késztermék befoglaló méretét, valamint bázist képzünk a többi forgácsoló megmunkálás számára. A lemarandó anyag magassága függ gyártandó termék méreteitől. Nagy előgyártmány esetében a felület egyenetlensége durvább, síklapúsága rosszabb lehet, míg kisméretű előgyártmányoknál ezen értékek csekélyebbek. A felhasználó beimportál egy előgyártmányt, vagy ennek hiányában olyan előgyártmányt feltételezünk, ami a késztermék befoglaló méreteitől minden irányban +3mm-rel nagyobb, de a fent említett viszonyok ismeretlensége miatt ez az érték az adatbázisban megváltoztatható. Homlokmarás esetében, mikor a felület teljes terjedelmét meg kell munkálni, rendszerint 45 fokos élszöggel ellátott marófejet alkalmaznak. Az eltávolítandó anyagmennyiség kiszámolható a sík felület befoglalóméretének és a "h" magasság szorzatából. Ebből időt a $t = \frac{V}{Q_t}$ összefüggésből kapunk, ahol Q_t az időegység alatt eltávolítható anyagmennyiséget jelöli. [23.]

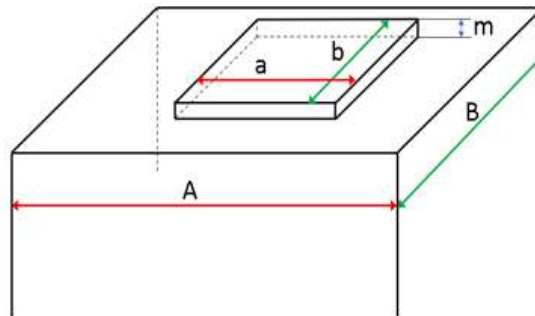
Sík felület palástmarása



38. Ábra: Palástmarás

Az előző pontban említett sík felületekre merőleges sík felületek megmunkálásához palástmarásra van szükség. Palástmarást szármaróval végzünk, ahol a radiális irányú fogásvétel sokkal kisebb, mint a rá merőleges axiális, ezért az eltávolítandó anyagvastagság kisebb. Alapértéknek a geometria méretbeli információ hiánya miatt a késztermék befoglaló méreteihez képest 3mm-rel nagyobb paramétert adtunk meg, ami az adatbázisban módosítható, vagy a felhasználó beimportál egy előgyártmányt A forgácsolandó anyagmennyiség a homlokmarással megegyező, de palástmarásnál “v” vastagsággal számolunk. Időt szintén a fenti összefüggéssel kalkulálhatunk. [23.]

Sík felület marása

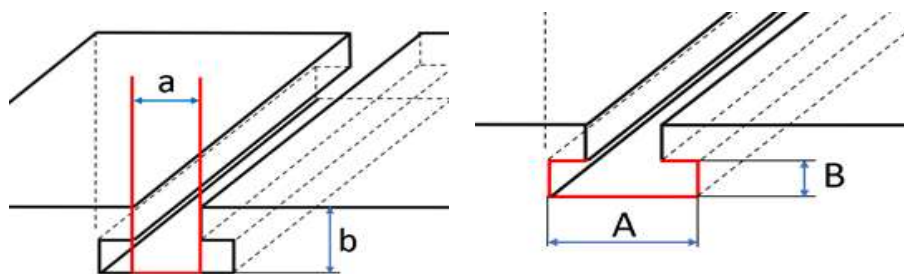


39. Ábra: Sík felület megmunkálása

Programunk által biztosított információkból tudjuk az alkatrész befoglaló méreteit, jelen esetben a test teljes szélességét és hosszát (A, B), továbbá a belső alak sajátosság mindhárom paraméterét (a, b, m). Ebből az információból megismerjük a belső elem magasságát (m). Szerszám kiválasztásakor figyelembe kell vennünk a belső feature pozícióját az adott felületen, ugyanis túlzottan nagy és kis megmunkáló eszköz választása a művelethez egyaránt hátrányos. Általánosan alkalmazott megoldás, hogy a gyártásért felelős szakember megméri a terven

szereplő feature modellhez legközelebb és legtávolabb eső távolságát, majd ezt átlagolva megkapja a forgácsoláshoz használandó szerszám méretét. Amennyiben ez nem esett pontos szerszámméretre, úgy a hozzá legközelebb eső nagyobb eszközt választja ki. Ezt a szintaktikát alkalmaztuk mi is a program megírásakor. A szerszám magával hordozza az adatbázisunkból a rá jellemző forgácsolási paramétereket, melyeket a technológiára jellemző összefüggés szerint alkalmazva megkapjuk az időegységenként eltávolítható anyagmennyiséget $[Q]$ -t. Ezt követően a befoglalóméret a $t = \frac{V}{Q_t}$ képletből áll elő, ahol V az eltávolítandó térfogat, míg Q_t az időegység alatt eltávolítható anyagmennyiséget jelöli. Az előgyártmány befoglaló mérete a késztermék befoglaló méreteinek növelésével kezelendő, melyre alapértéket adtunk meg, ami szabadon változtatható. [23.]

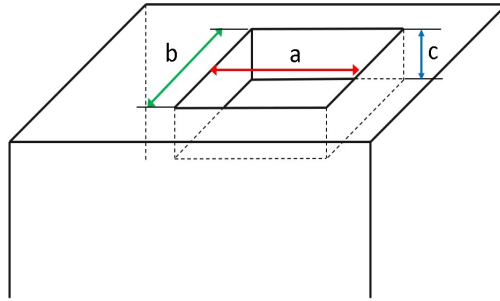
Hornyok marása (pl.: T-horony)



40. Ábra: T-horony első és második lépés

T alakú hornyok marására a termék, vagy a termékhez való rögzítés, illetve megvezetés céljából van igény. Elkészítése két lépcsőből áll. Első lépésben a horony keskenyebbik részét (a) marjuk a végleges mélységig (b) ujjmaróval. Itt célszerű a horony szélességével megegyező átmérőjű marót választani. Második lépésben a T-alak megmunkálásához T horonymarót használunk, amivel kialakítjuk az “A” szélességű és “B” magasságú alakot. Amennyiben a horony “B” magassága nagyobb, mint a marófeje, úgy több fogást eszközölünk. A teljes alaksajátosság számítására a két lépést átfedés nélkül egyesítjük, így a “T” alak felülete az $a * (b - B) + A * B$ képlettel megadható. Ezt a horony teljes hosszára kiterjesztve megkapjuk térfogatot $[V]$. A feature előállítási ideje a $t = \frac{V}{Q_t}$ összefüggéssel megadható. [23.]

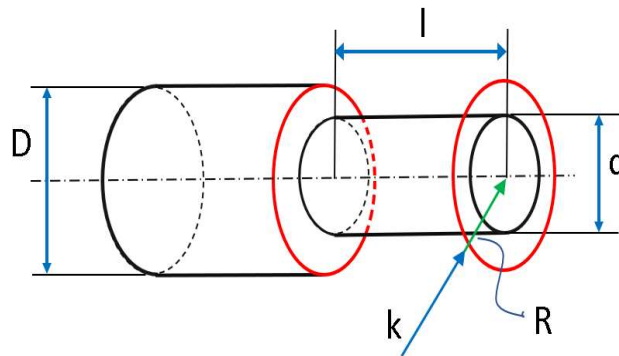
Zsebek marása



41. Ábra: Zseb kialakítása

Zsebek megmunkálására leginkább az alakkal záró kötések kialakítása miatt van szükség. Ezek forgácsolása ujjmaróval történik és a szerszám kiválasztása a már említett megoldás szerint zajlik. A belső lekerekítési sugár figyelembevételével választható a sugárhoz igazodó szármaró is, ugyanakkor nagy méretű zsebek marásakor fennáll a lehetőség, hogy a kis lekerekítési sugár miatt alkalmazott maróval nagyságrendileg több ideig tart a megmunkálás. Az eltávolítandó anyag a zseb paraméterivel kalkulálható általános térfogatszámítási megoldással ($a * b * c$) alakban. A megmunkálási idő a szokásos $t = \frac{V}{qt}$ összefüggéssel megadható. [23.]

Hossz- és kereszttergálás

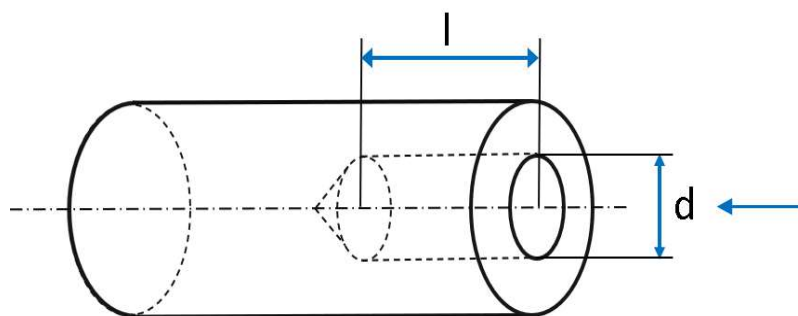


42. Ábra: Esztergálás

Esztergálással forgásszimmetrikus alkatrészeket gyártunk, melyek tengelyek, tárcsák formájában vannak jelen. Esztergálásnak több különböző fajtája van, de mi most az oldalazásra és hosszesztergálásra fókuszálunk. Első lépésben az oldalazást végezzük el, ahol a szerszám a munkadarab tengelyére merőleges halad “k” iránnyal. A forgácsolás a legnagyobb átmérőn (D) kezdődik és a tengely középpontjában végződik, ami “R” megtett útnak felel meg ($R = \frac{D}{2}$).

Oldalazásnál az eltávolított anyagmennyiség számolható a $\frac{D^2\pi}{4} * ap$ képlettel, ahol ap az oldalazáskor vett fogásmélység. Hosszesztergálásnál a szerszám párhuzamosan halad a forgástengellyel. Itt a forgácsolás szintén a legnagyobb (D) átmérőn kezdődik és a végső (d) átmérőn fejeződik be az esztergálás teljes hosszában (l). A fogásmélység adatbázisból adódik. Eltávolítandó anyagmennyiség a $(\frac{D^2\pi}{4} - \frac{d^2\pi}{4}) * l$ összefüggéssel kalkulálható. A megmunkálási idő pedig $t = \frac{V}{q_t}$ képletből számolható. [23.]

Fúrás

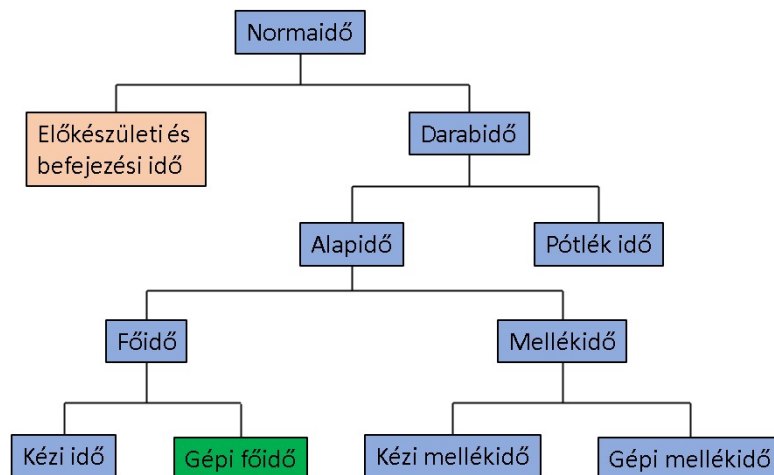


43. Ábra: Fúrás

Fúrást alkalmazunk kikönnnyítésre, előzetes megmunkálásra, rögzítési lehetőségre, megfogási pontra. Ezen megmunkáláskor a fúroszerszám tengelye a munkadarab tengelyével megegyező, a megmunkált átmérő azonos a szerszám átmérőjével. Az eltávolított anyagmennyiség megadható a furat felületének és a furat hosszának szorzatából. $(\frac{d^2\pi}{4} * l)$. A megmunkálás ideje a már említett $t = \frac{V}{q_t}$ összefüggésből adódik. [23.]

VI.2 Normaidő

A megmunkálási időt több gyártási tényező kifejezésére lehet felhasználni, mint ütemezésre, kapacitás előrejelzésre, hogy a gyártó mennyi időre tud előre vállalni, vagy mint ahogyan mi is a kezdeti célunknak megfelelően gyártási költség becslésre. A költségbecslés előtt azonban összegeznünk kell a teljes gyártási időbecslést, ami tulajdonképpen a normaidő meghatározása, melyet az alábbi ábra szemléltet.



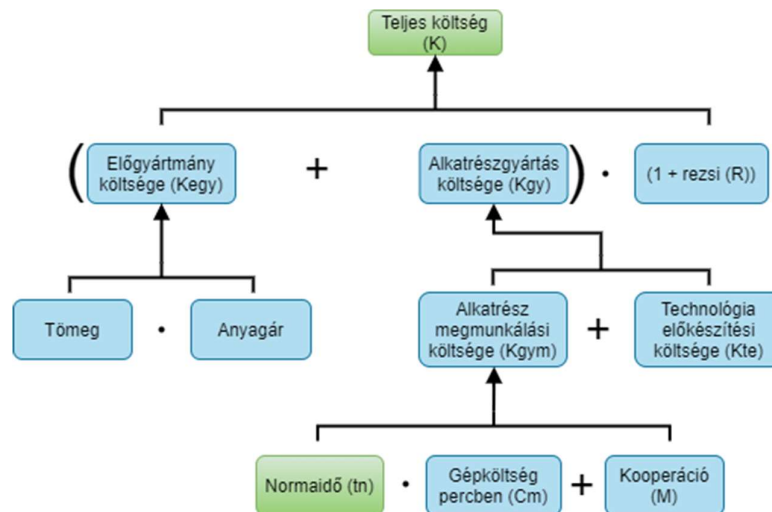
44. Ábra: Normaidő meghatározása

A mi predikciós eljárásunkkal a zölddel jelzett gépi főidőt vagyunk képesek becsülni, míg kézi főidőnek nem tekintünk jelentőséget a gyártás automatizált mivolta miatt. A kézi mellékidő esetenként lehet mérvadó adat, amelyet kétlépcsős értékként kezelünk, azaz egy adott tömeg felett, amit már daruval kell mozgatni és azon tömeg alatt amikor elegendő a kézi anyagmozgatás. E két értéknek szintén szabadon állítható értéket hagytunk, de alapértelmezett értékünk 0. A kapott főidőnk, ami egyben a gépi főidőt jelenti a gépi alapidő döntő hányada. A mellékidő a főidő egy bizonyos százaléka, mely adatbázisban beállítható. Kézi mellékidő alatt a munkadarab ki- és befogási idejét értjük, míg gépi mellékidőnek a szerszámgép nem forgácsolással töltött idejét tekintjük. Egy munkadarab legyártását darabidőnek értelmezzük, ami alapidőből és pótlék időből áll, ahol az alapidőt az eddig megnevezett idők összegzéséből kapjuk és a pótléki idő rendszerint az alapidő egy bizonyos százalékos értéke, mely 10% alatt van. Utóbbi magában foglalja a gépkezelő kiegészítő tevékenységekre fordított idejét, mint forgácseltávolítás, vagy szociális szükségletek. A darabidő ismeretében már csak az előkészületi és befejezési időt kell tisztáznunk a normaidő megadásához. Előkészületi és befejezési időnek tekinthető például a szerszám beillesztése, vagy az eredeti megmunkálási előtti állapot visszaállítása. Ennek ismeretére bekérjük a gyártótól az információt, de addig is megállapítunk egy hozzávetőleges időt.

VI.3 Költségbecslés

VI.3.1 Költségbecslés a gyakorlatban

A költségbecslés folyamatát több tényező befolyásolja, melyeket egyszerűsítve, de a korrekt értéket megtartva létrehozhatjuk a következő ábrát az átláthatóság érdekében.



45. Ábra: Költségmeghatározás folyamata

A költségbecsléshez szükséges normatív adatok adatbázisban vannak eltárolva, melyek módosítására lehetőséget kínálunk. A teljes költséget három részből kapjuk meg.

Az előgyártmány költségét meghatározhatjuk a gyártandó alkatrész tömegének és az aktuális alapanyagárnak a szorzatából, ahol az alkatrész tömegét ismerjük a STEP fájlból, az anyagár pedig adatbázisbeli érték, mely folyamatos frissítést igényel.

A “Kgy” mint alkatrészgyártás költség további két részköltségből tevődik össze. Itt beszélhetünk csupán a termék megmunkálási költségéről (Kgym), ami előáll, ha a megmunkálás normaidejét megszorozzuk a gép percköltségével, ezt elvégezzük az összes műveletelemre majd hozzáadjuk a kooperáció költségét. Kooperáció nem minden alkatrész gyártásakor fordul elő, de ennek megadására lehetőséget biztosítunk. A technológiai előkészítési költség biztosít egy plusz költséget, ami valamilyen kiugró eszköz árára érthető. [25.] [26.]

Végül szerepel egy szorzó tényező is az összefüggésben, ahol az “R” mint általános vállalati rezszi költség van nyilvántartva. A rezszi (R) vállalatonként változó érték, így ezt is újra definiálható értéként kezeljük. [25.]

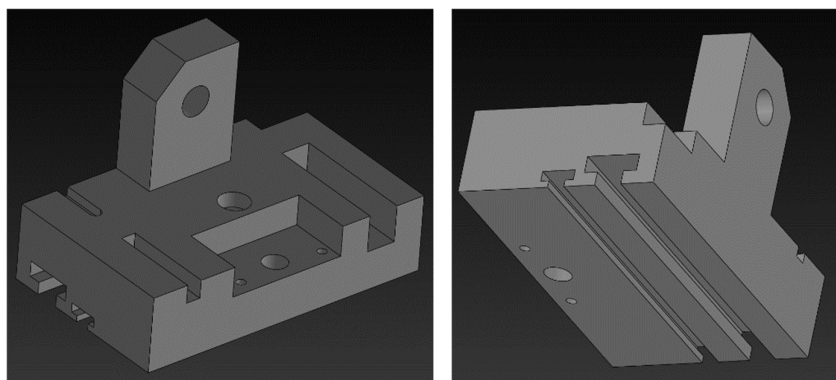
VII Mintapélda

Ebben a fejezetben sor kerül az elkészült szoftver működésének áttekintésére, amely során egy forgásszimmetrikus és egy prizmatikus alkatrészt veszünk górcső alá. A felhasználó számára az user interfacen lehetőség van az előre definiált szerszám és gépi információkat tartalmazó adatbázis bővítésére, lásd 46. Ábra. Továbbá a munkadarab STEP fájlján felül a kívánt előgyártmányt is betölthető, abban az esetben, ha erre nem kerül sor az algoritmusunk a befoglaló méretek segítségével definiálja azt. A STEP fájlok beimportálása után pedig indulhat is a procedúra.

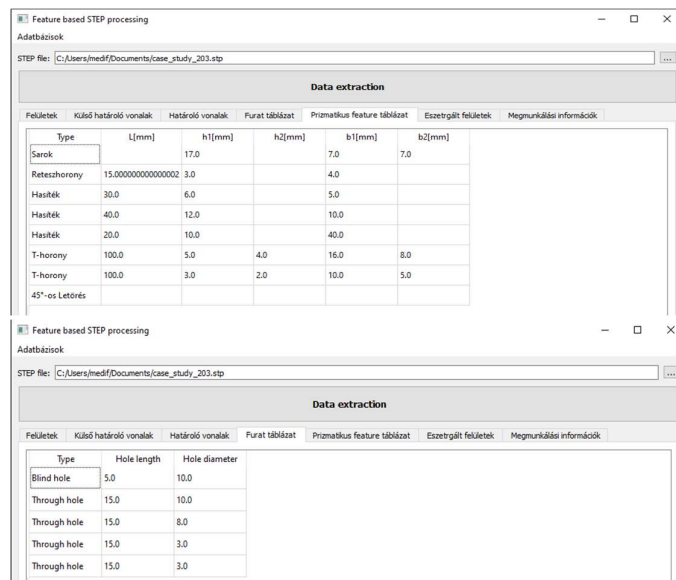
Megmunkálás jellege	Szerszám típusa	Átmérő D [mm]	Forgácsolási él sz. sz [db]	Fogankénti előtolás fn [mm]	Fordulatszám n [min-1]	Forgácsolási szélesség ae [mm]	Fogásmélység ap [mm]
marás 45	R245-032A32-12M	32.0	3.0	0.354	3520.0	25.0	6.0
marás 45	R245-040A32-12L	40.0	3.0	0.354	2870.0	31.25	6.0
marás 45	R245-050A32-12L	50.0	3.0	0.354	2340.0	38.46	6.0
marás 45	R245-063A32-12L	63.0	4.0	0.354	1850.0	50.0	6.0
marás 45	R245-080Q27-12L	80.0	8.0	0.354	1380.0	62.5	6.0
marás 45	R245-100Q32-12H	100.0	10.0	0.354	1170.0	71.43	6.0
marás 45	R245-125Q40-12H	125.0	12.0	0.354	898.0	100.0	6.0
marás 45	R245-160Q40-12H	160.0	16.0	0.354	641.0	125.0	6.0
marás 45	R245-080Q32-18H	80.0	5.0	0.283	1420.0	62.5	10.0
marás 45	R245-100Q32-18H	100.0	6.0	0.307	1360.0	35.71	10.0
marás 45	R245-125Q40-18H	125.0	7.0	0.283	909.0	100.0	10.0
marás 45	R245-160Q40-18H	160.0	9.0	0.283	696.0	125.0	10.0
marás 90	490-020A16-08L	20.0	2.0	0.2	5720.0	17.86	5.5
marás 90	490-025A20-08L	25.0	2.0	0.2	4580.0	22.73	5.5
marás 90	490-032A25-08L	32.0	3.0	0.2	3670.0	29.41	5.5
marás 90	490-040A32-08L	40.0	6.0	0.21	2570.0	35.71	5.5
marás 90	490-050C5-08H	50.0	7.0	0.21	2350.0	45.45	5.5
marás 90	490-063C6-08H	63.0	8.0	0.21	2000.0	55.56	5.5

46. Ábra: A bővíthető adatbázis

Az alábbi, 47. Ábrán a megvizsgált prizmatikus alkatrész található. Míg a 48. Ábrán a kinyert prizmatikus alaksajátosságok és a furattábla.

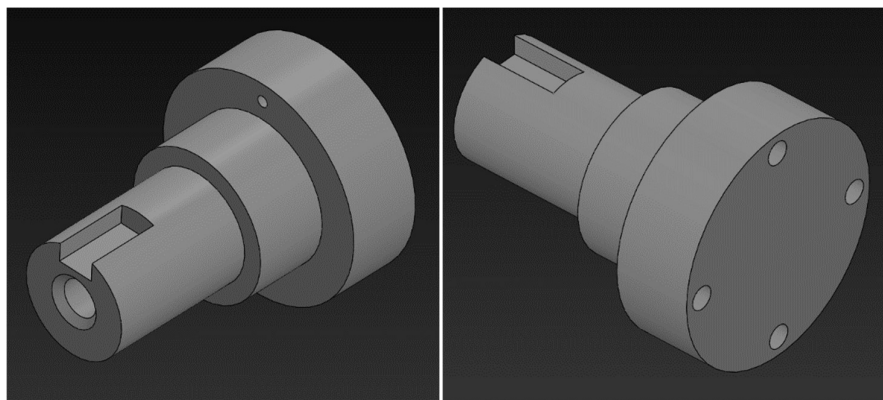


47. Ábra: A vizsgált prizmatikus alkatrész

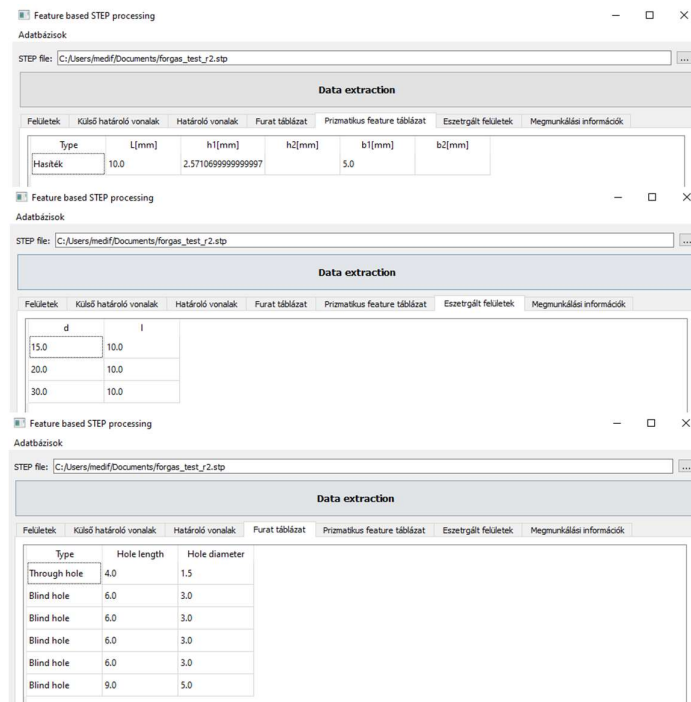


48. Ábra: A prizmatikus alkatrészhez tartozó alaksajátosságok

Az alábbi, 49. Ábrán pedig a vizsgált forgásszimmetrikus alkatrész található, alatta az xy ábrán pedig a hozzá köthető alaksajátosságok.



49. Ábra: A vizsgált forgásszimmetrikus alkatrész



50. Ábra: A forgásszimmetrikus alkatrészhez tartozó alaksajátosságok

Az alaksajátosságok megtalálása után becsülhető a várható megmunkálási idő majd abból a költség. A “Start prediction” gomb megnyomása után a költségszámításhoz kapcsolódó értékeket kell áttekinteni és tetszés szerint változtatni, lásd 51. Ábra.

Start prediction

Adatbekérő

Kézi fűző Default: 4

Kézi mellékfűző Default: 60

Gépi mellékfűző [%] Default: 0.2

Pótlékfűző [%] Default: 0.1

Előkészületi és befejezési idő Default: 720

Gépköltség [min] Default: 250

Kooperáció Default: 0

Anvao Default: C25

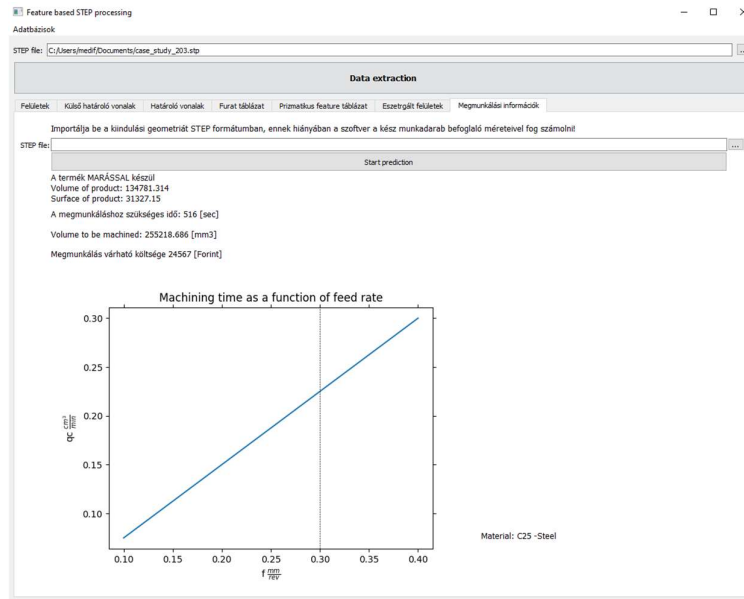
Anvaór [Ft/ka] Default: 1500

Technológiai előkészítés költsége [Ft] Default: 0

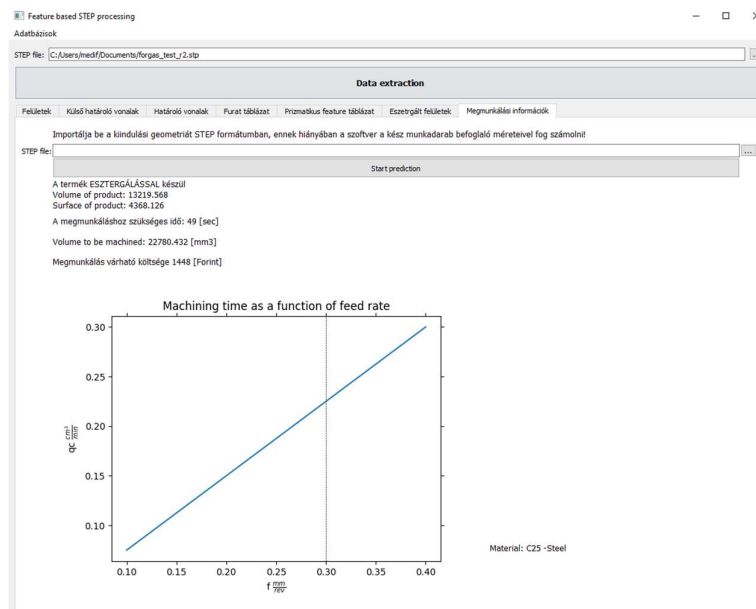
Rezsi [Ft/min] Default: 0,004

Mentés

51. Ábra: A költségszámításhoz szükséges értékek

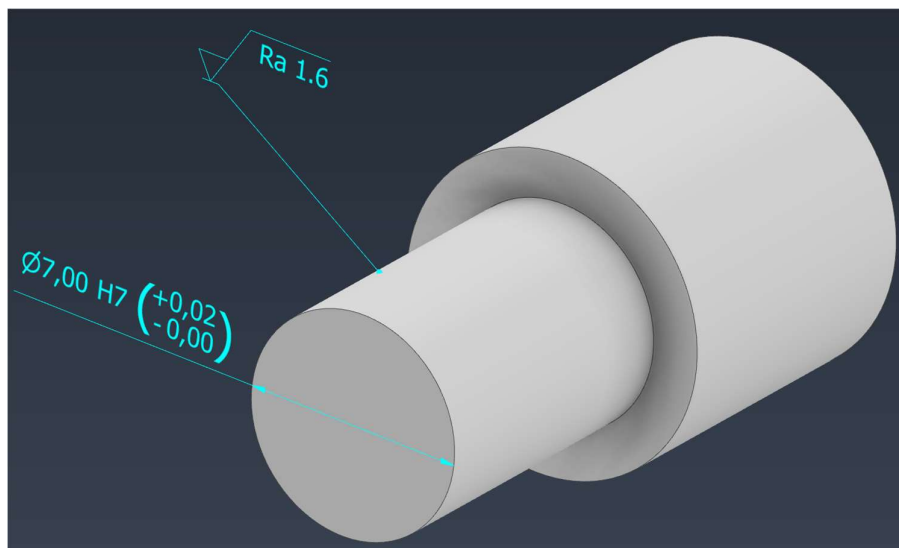


52. Ábra: A prizmatikus alkatrész marással történő megmunkálásának jellemzői



53. Ábra: A forgásszimmetrikus alkatrész esztergálással történő megmunkálásának jellemzői

A dashboardon, az 52. és 53. Ábrán egyelőre egy kezdetleges diagram, valamint elrendezés látható, viszont a becsült értékek leolvashatóak és a lényegi koncepció is látható. Az eredményekkel elégedettek vagyunk.



54. Ábra: Finommegmunkálás

	Köszörülés	Simítás	Nagyolás
IT	< IT7	IT7 - IT11	> IT11
Ra	< 1,6	1,6 - 12,5	>12,5

55. Ábra: Méretpontosság határok

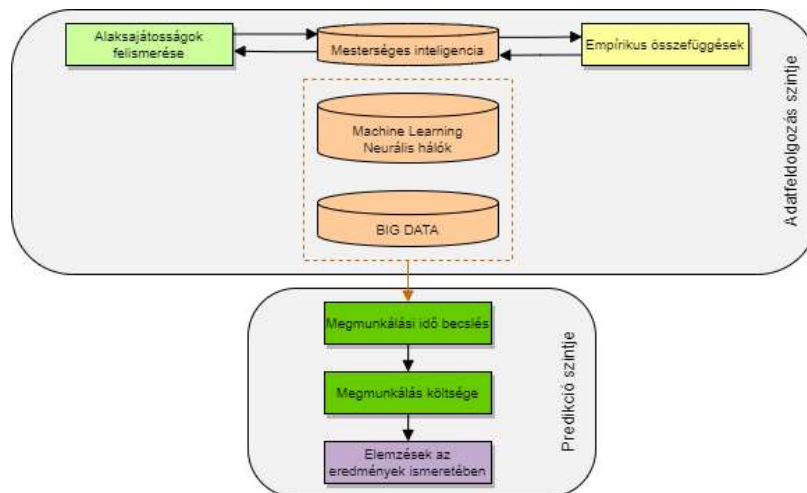
A méretpontosságokat és felületi érdességeket úgy vettük figyelembe, hogy egy adatbázisban előre eltárolt határértékeket vizsgáltunk. Egy megadott határérték alatt előírtuk a köszörülést, a két határérték között előírtuk a simítást, ezek normaidejét felületarányosan meghatároztuk, egy bizonyos határérték felett viszont kizárólag nagyolást alkalmaztunk.

VIII Konklúzió és jövőbeli fejlesztési lehetőségek

A dolgozatra a tesztek elvégzése után egészként tudunk tekinteni, látszik, hogy az elgondolások működnek gyakorlatban is. A STEP fájlokból sikeresen tudjuk kinyerni az alaksajátosságok felépítéséhez szükséges információkat, amelyekből képesek vagyunk kategóriába sorolni az adott featuret, majd társítani a hozzá tartozó geometriai jellemzőket. Utóbbiak tudatában a megmunkálás típusának megállapítása is megtörténik, amely után az előre definiált adatbázisok és számítási összefüggések segítségével a megmunkálandó térfogat mennyiség függvényében gyártási időt becsülhetünk. Jelen dolgozatban az idő és az ez által definiált költség került meghatározásra, azonban tekintve, hogy a gyártásban az idő mi mindenre felhasználható, hisszük, hogy ez egy előremutató tématerület. A dolgozat kiméretében idáig szerettünk volna eljutni, az eredményeket kielégítőnek találjuk.

Azonban abból kifolyólag, hogy ez egy igen progresszív ága az ipari automatizáció világának, talán nem meglepő, hogy a sok elért eredmény mellett számos olyan fejlesztési lehetőséget vízionálunk, melyek a megírt szoftver és az alkalmazott módszerek elengedhetetlen részeivé válhatnak.

Az első és legfontosabb fejlesztés, hogy a programunk univerzálisan minden STEP fájlt kezelni tudjon. Ismerve a formátum teljes felépítését, ez egy igen hosszú feladatnak ígérkezik. A második terület, amellyel olyan mélyen nem foglalkoztunk, hogy bekerüljön a dolgozatba az az egymást többszörösen metsző alaksajátosságok kérdése. A programunk ezen tekintetben limitált, hiszen a kombinációjuk végtelen. Ha a programunk képes különálló feature-öket lekezelni az egymást metszőkön belül az nem jelent jelentős problémát, viszont, ha emiatt a működése kiszámíthatatlanná válik, azzal érdemes foglalkozni. Ez abból ered, hogy jelenleg a leggyakrabban előforduló alaksajátosságok felismeréséhez szükséges előre létrehozott kategóriákkal dolgozunk. Belátható, hogy léteznek, olyan bonyolultabb, kategóriákba nem sorolható alaksajátosságok, amelyekkel nem boldogulunk. Ezen alaksajátosságok, és összességében a végtelen mennyiségű kategória létrehozásának elkerülésére egy lehetséges választ kínál az 55. Ábra.



55. Ábra: A jövőbeli célkitűzéseink

Amint látható, a mesterséges intelligencia területével még nem foglalkoztunk, azonban a szoftver felépítése tudatosan előkészítette az implementálásának lehetőségét, többek között ezért is íródott Python környezetben. A jövőben elsődleges célunk a kategorizálás mesterséges neurális hálók használatával megvalósított gépi tanulási algoritmusok segítségével történő kiváltása. Így a szoftverünk képessé fog válni, bármilyen bonyolultságú és egymást metsző alaksajátosságok felismerésére.

Továbbá itt térhetünk rá a dolgozat második részének pontatlanságaira. Az általunk használt tapasztalati modell jelenleg minimálisan adaptív, és nem vesz számításba külső befolyásoló tényezőket. Így habár az eredményül kapott értékek jó kiindulást nyújthatnak, nagyon pontos előrejelzés szempontjából nem validálhatóak, ezáltal nem tekinthetők mérvadónak. Viszont itt jön képbe az úgy nevezett Big Data elemzések implementálhatósága. Amennyiben a felhasználó rendelkezik a gyártási körülményeit jellemző korábbi megtérülési értékekkel és ezek száma elég nagy, akkor képesek vagyunk egy valós tapasztalati modell felállítására, amely hosszú távon önvalidációra is alkalmas lehet. Végeredményképp pedig egy olyan predikció jöhet létre, amely teljesen felhasználó függően lenne képes adaptálódni és közvetve a külső befolyásoló körülményeket is figyelembe venné, ezáltal egy valós ipari megoldást kínálva a felhasználónak.

A tisztán látás végett, úgy gondoljuk, hogy igen nagy fába vágtuk a fejszénket, ebből is következnek a tárgyalt hiányosságok. Azonban a tudatosan felépített struktúrának köszönhetően a jelenlegi munkánk egy nagyon erős és használható alapot kínál a felhasználók számára, amely jelentős potenciállal rendelkezik az Ipar 4.0 megoldásainak témakörében.

IX Irodalomjegyzék

- [1.] „STEP File Format (CAD) – Simply Explained”. 2020. All3DP. 2020. november 29.
<https://all3dp.com>
- [2.] Nguyen, Jimmy. „Best STEP File to Use: AP203, AP214, and AP242”. Elérés 2021. szeptember 15. <https://www.capvidia.com/blog/best-step-file-to-use-ap203-vs-ap214-vs-ap242>.
- [3.] „STEP ARM Model”. é. n. Elérés 2021. november 7.
http://www.steptools.com/stds/stp_expg/arm.html.
- [4.] Bhandarkar, Mangesh P., és Rakesh Nagi. „STEP-Based Feature Extraction from STEP Geometry for Agile Manufacturing”. Computers in Industry 41, sz. 1 (2000. január): 3–24.
- [5.] Sreeramulu D, és CSP Rao. „A New Methodology for Recognition of Milling Features from STEP File” 6, sz. 3 (é. n.): 19.
- [6.] Malleswari, V N. „Automatic Recognition of Machining Features Using STEP Files”. International Journal of Engineering Research 2, sz. 3 (2013): 11.
- [7.] Nasr, Emad Abouel, és Ali K. Kamrani. Computer-Based Design and Manufacturing: An Information-Based Approach. New York: Springer, 2007.
- [8.] Venu, Bitla, és Venkateswara Rao Komma. „STEP-Based Feature Recognition from Solid Models Having Non-Planar Surfaces”. International Journal of Computer Integrated Manufacturing 30, sz. 10 (2017. október 3.): 1011–28.
- [9.] Srivastava, Deepanshu, és Venkateswara Rao Komma. „Development of STEP AP224 Extractor for Interfacing Feature Based CAPP to STEP-NC (AP238)”. International Journal of Automation and Computing 16, sz. 5 (2019. október): 655–70.
- [10.] J. Shah, M. Rogers, Functional requirements and conceptual design of the feature-based modeling system, Computer Aided Engineering Journal 5
- [11.] Shanthakumar, Aravind. „Development of Feature Recognition Algorithm for Automated Identification of Duplicate Geometries in CAD Models”, é. n., 161.
- [12.] Joshi, S., és T.C. Chang. 1988. „Graph-Based Heuristics for Recognition of Machined Features from a 3D Solid Model”. Computer-Aided Design 20 (2): 58–66.
- [13.] De Floriani, L. 1987. „A Graph Based Approach to Object Feature Recognition”. In Proceedings of the Third Annual Symposium on Computational Geometry - SCG '87, 100–109. Waterloo, Ontario, Canada: ACM Press.

- [14.] Vandenbrande, J.H., és A.A.G. Requicha. 1993. „Spatial Reasoning for the Automatic Recognition of Machinable Features in Solid Models”. IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (12): 1269–85.
- [15.] Babic, Bojan, Nenad Nesic, és Zoran Miljkovic. 2008. „A Review of Automated Feature Recognition with Rule-Based Pattern Recognition”. Computers in Industry 59 (4): 321–37.
- [16.] Sakurai, Hiroshi. 1995. „Volume Decomposition and Feature Recognition: Part 1—Polyhedral Objects”. Computer-Aided Design 27 (11): 833–43.
- [17.] Gao, S, és J J Shah. é. n. „Automatic Recognition of Interacting Machining Features Based on Minimal Condition Subgraph”, 13.
- [18.] Rameshbabu, V., és M.S. Shunmugam. „Hybrid Feature Recognition Method for Setup Planning from STEP AP-203”. Robotics and Computer-Integrated Manufacturing 25, sz. 2 (2009. április)
- [19.] Cai, Na, Soumiya Bendjebba, Sylvain Lavernhe, Charyar Mehdi-Souzani, és Nabil Anwer. „Freeform Machining Feature Recognition with Manufacturability Analysis”. Procedia CIRP 72 (2018):1475–80.
- [20.] Friedl, Jeffrey E. F. Mastering Regular Expressions. 3rd ed. Sebastapol, CA: O’Reilly, 2006.
- [21.] Borkar B.R., Puri ,Kuthe A. M., Deshpande P. S. “Optimization of Machining Time using Feature Based Process Planning” Elérés 2021. Október 21
- [22.] Richard Crowson, “Factory Operation Planning and Instructional Methods” Elérés 2021. November 16.
- [23.] Eckerle Group, “Mechatronic components for mobility and comfort” cég Látogatás 2021. Október 22.
- [24.] Mikó Balázs, Technológiai előtervezés automatizálása mesterséges intelligencia módszerek segítségével, Elérés ideje: 2021. Október 24.
- [25.] Dr. Mikó Balázs, Lemezalakító szerszámok költségbeclése, Elérés ideje: 2021. Október 30.
- [26.] Han, JungHyun, Mujin Kang, és Hoogon Choi. 2001. „STEP-Based Feature Recognition for Manufacturing Cost Optimization”. Computer-Aided Design 33 (9): 671–86.