



Inverz posztprocesszor fejlesztése a CAD/CAM/CNC munkafolyamat rugalmasságának növelésére

Development of an inverse postprocessor to increase the flexibility of the CAD / CAM / CNC workflow

¹Bacs József és ²Geier Norbert

¹*Budapesti Műszaki és Gazdaságtudományi Egyetem, Gépészmérnöki Kar, Gyártástudomány és -technológia Tanszék. Budapest, Magyarország, bacsjoz012@gmail.com*

²*Budapesti Műszaki és Gazdaságtudományi Egyetem, Gépészmérnöki Kar, Gyártástudomány és -technológia Tanszék. Budapest, Magyarország, geier@manuf.bme.hu*

Összefoglalás

Ipari környezetben gyakran előforduló nehézséget jelent egy már nem elérhető programozó(rendszer) által írt NC program illesztése valamely, az eredeti programtól jelentősen eltérő logikájú és szintaktikájú vezérlésre. A kutatásunk fő célja egy inverz posztprocesszor kifejlesztése volt, mely algoritmus egy meglévő NC programot szabványos CLDATA nyelvre fordít vissza. A ZW3D szoftver és az ISO 3592:2000 szabvány alapján fejlesztettünk egy rendszerfüggetlen algoritmust NC program inverz posztprocesszállására. Ezen algoritmus alapján C# környezetben felhasználóbarát programot hoztunk létre, melyet 2D-s esztérgálási és 2,5D-s marási környezetekben teszteltünk. A kifejlesztett algoritmus időhatékonyan, rugalmasan és biztonságosan használható —a specifikációkban rögzített feltételek mellett— NC program CLDATA programmá való visszaalakítására.

Kulcsszavak: CAD/CAM/CNC, posztprocesszor, CLDATA, inverz posztprocesszor

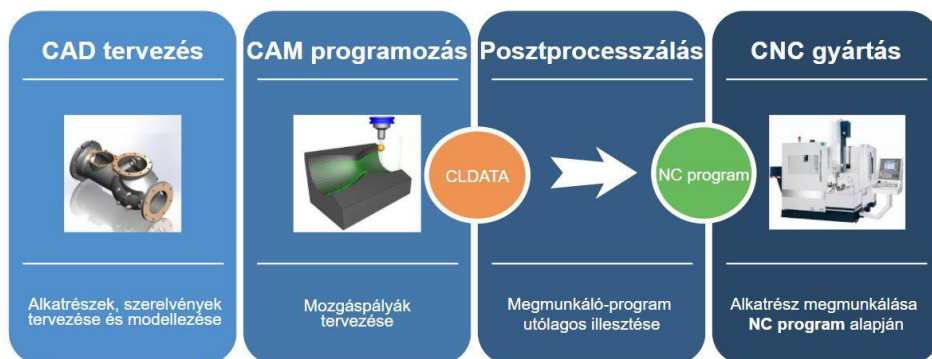
Abstract

In an industrial environment, it is often difficult to fit an extant NC program to that other controller with logic and syntax differ significantly from the original one. The main goal of our research was to develop an inverse postprocessor, which algorithm translates an existing NC program back to CLDATA. Based on the ZW3D software and the ISO 3592:2000 standard, we developed a system-independent algorithm for the inverse postprocessing of an NC program. Based on this algorithm, we developed a user-friendly program in C#, which was tested in 2D turning and 2.5D milling environments. The developed algorithm can be applied time-efficiently, flexibly, and safely under the conditions specified in the specifications, to convert an NC program back to CLDATA.

Keywords: CAD/CAM/CNC, postprocessor, CLDATA, inverse postprocessor

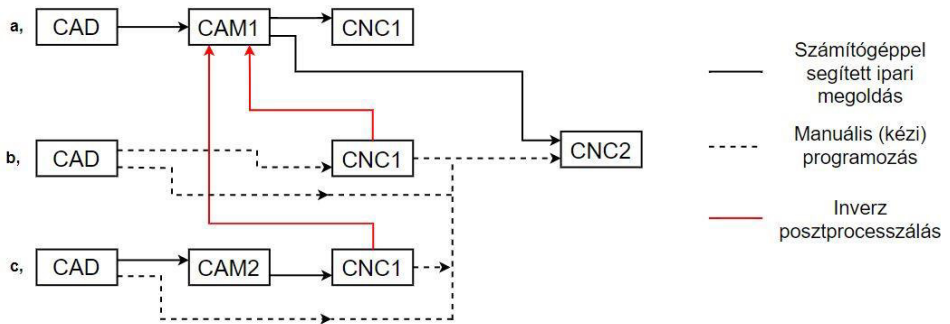
1 Bevezetés

A bonyolult geometriájú forgácsolt alkatrészeket jellemzően CNC szerszámgépeken gyártják, amennyiben a gyártás ütemezése és/vagy tömegszerűsége ezt indokoltá teszi. Ezen alkatrészek gyártóprogramját leggyakrabban különböző számítógéppel segített CAD/CAM tervezőrendszerekkel állítják elő [1]. Ezen széleskörűen elterjedt és alkalmazott ipari munkafolyamatot szemlélteti az 1. ábra.



1. ábra: CAD/CAM/CNC munkafolyamat

Az alkatrészek 3D-s testmodelljeinek elkészítése után (amit jellemzően egy konstruktőr végez) lehetőségünk nyílik egy CAM szoftver segítségével megmunkálást programozni. A programozott geometriai és technológiai információk ISO 3592:2000 szabványos CLDATA nyelven tárolódnak a CAM rendszerekben, sokszor a felhasználó számára nem elérhető formában. A CLDATA olvasható szövegszerkezetű (ASCII kódos) programnyelv, alapvető egysége a rekord (mondat) [2]. A CLDATA adatokat a megmunkálásra kiválasztott szerszámgép vezérlőjének nyelvére a posztprocesszor segítségével tudjuk utólag illeszteni (fordítani) [3]. Az utólagos illesztéssel a megmunkálásra felhasznált vezérlő környezetében megkapjuk azt az NC programot, amely alapján az alkatrészgyártás megtörténik [1, 4]. Ezt az iparban gyakran előforduló folyamatot szemlélteti a 2.a ábra. Szintén gyakran előforduló eset egy forgácsolt alkatrészeket gyártó vállalatnál, hogy egy már elkészült 3D-s CAD modell vagy alkatrészrajz alapján kézi programírással hozzák létre az NC programot az adott vezérlő környezetében, mely folyamatot a 2.b ábra szemléltet. Előfordul azonban olyan eset is, hogy az NC programot egy olyan szoftver segítségével generálták, amelyhez már nincsen hozzáférésük (pl. érvényes licenz vagy működőképes hardver) (2.c ábra), vagy helyette egy más/újabb szoftvert használunk. A gyártás során gyakran előforduló feladat, hogy egy másik típusú vezérlésre kell áthelyezni egyes gyártási tevékenységeket (pl.: újraütemezés, tervezett karbantartás, meghibásodás miatt), ahogy azt az ábrán „CNC2” szerszámgép szemlélteti.



2. ábra: Az inverz posztprocesszor szükségessége

A 2.a ábrán szemléltetett esetben (rendelkezésünkre áll a gyártóprogram CLDATA formátumban) a gyártás átütemezése egy másik vezérlőjű szerszámgépre egyszerűen és gyorsan véghezvihető. A 2.b és 2.c ábrán szemléltetett esetekben azonban kevésbé rugalmas az átállítás. A kutatómunkánk fő célja egy inverz posztprocesszor fejlesztése volt, mellyel a 2. ábrán piros nyilakkal szemléltetett módon egy meglévő NC program visszaalakítható szabványos CLDATA programnyelvre, ezáltal kiküszöbölve az időigényesebb és nehezebb manuális megoldásokat (szaggatott vonal), és új kapcsolatokat hozva létre, amelyek elősegítik a már meglévő számítógéppel támogatott, gyors és rugalmas ipari folyamatokat (folytonos fekete vonal).

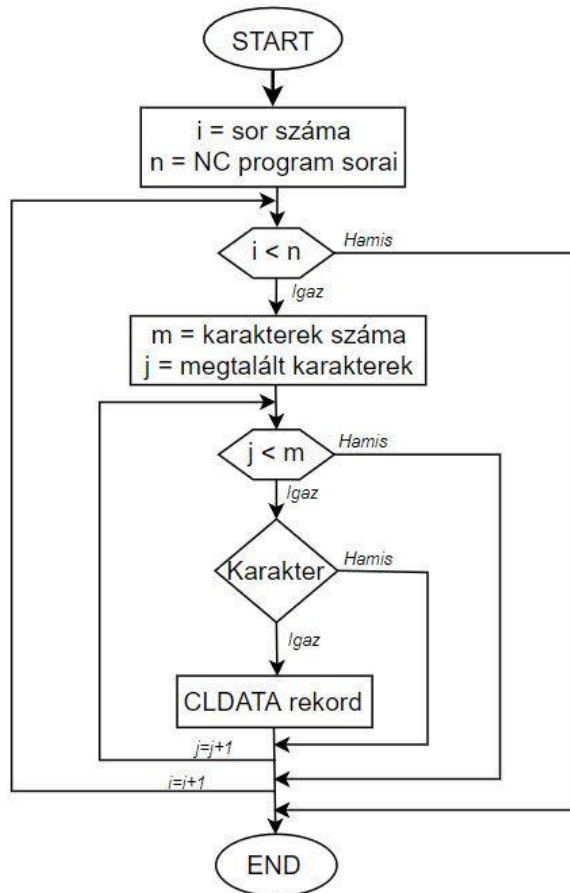
2 Követelményjegyzék

Egy követelményjegyzékben foglaltuk össze azokat a főbb szempontokat, amelyeket figyelembe vettünk a fejlesztői munkában: (i) az általunk fejlesztett inverz posztprocesszor alkalmas Fanuc és Milltronics 2,5D-s marógépekre, és 2D-s esztergagépekre készített NC programok kezelésére; (ii) az inverz posztprocesszor alkalmazása lehetővé teszi alapvető megmunkálások, kontúrsimítás, síkmarás, valamint az ezekhez szükséges műveletelemek NC programjainak visszaalakítását szabványos CLDATA nyelvre; (iii) az inverz posztprocesszor képes visszaalakítani az NC mondatban szereplő szerszámszámra, fordulatszámra, forgásirányra, és előtolásra vonatkozó parancsokat, valamint a szerszámsugárkorrekcióra, abszolút koordinátamegadásra, és különböző M-parancsokra (M05, M08, M09, M30) vonatkozó utasításokat; (iv) képes továbbá az inverz posztprocesszor a gyorsmeneti pozicionálást és lineáris interpolációs mozgásokat, az adott koordináta értékekkel szabványos CLDATA formátumba visszaalakítani.

3 Az inverz posztprocesszor algoritmus

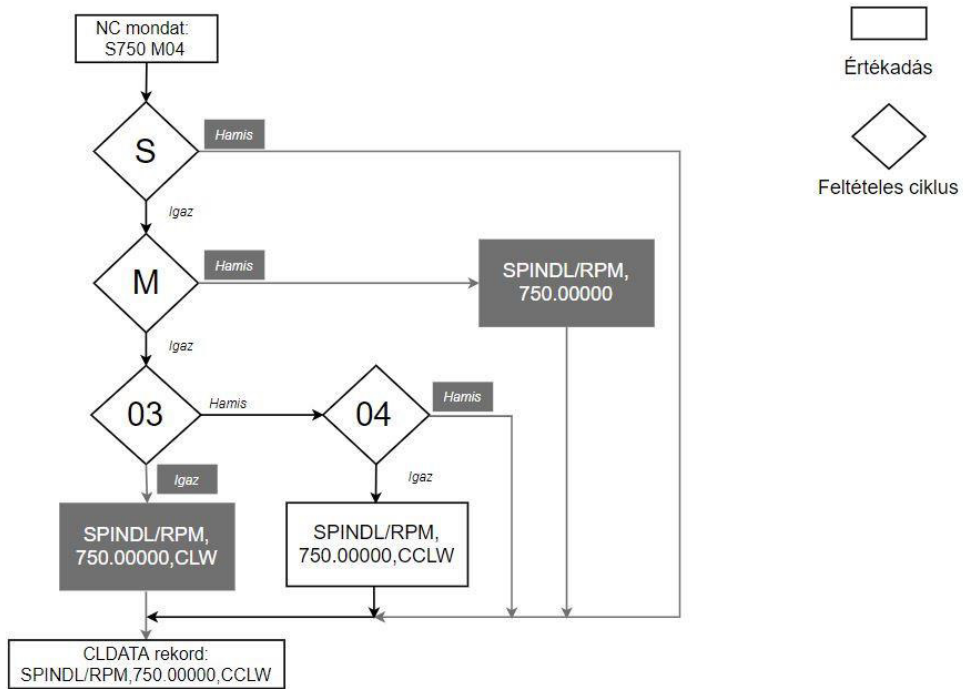
Az általunk kifejlesztett algoritmus feltételes és iterációs utasításokból épül fel, bemenetként az NC program szavait vizsgálja, kimenetei pedig a CLDATA rekordok. Az algoritmus első lépésben az NC programot úgy alakítja át, hogy azon tartalmi változást nem alkalmaz, azonban formailag a CLDATA szerkezetéhez szabja (pl.: G1 X_x Y_y Z_z, ahol x,y,z az interpolált mozgásciklus végkoordinátái). Az így kapott, kiegészített NC programon a C# környezetben leprogramozott algoritmust lefuttatva visszakapható az eredeti CLDATA program. Az általunk fejlesztett algoritmus átfogó bemutatása a 3. ábrán látható. Az algoritmus minden egyes keresett parancs esetén addig fut, míg az az NC programban fellelhető.

Az elkészült algoritmussal az NC programban az adott parancsra jellemző karakter megkeresésével visszakaphatók szabványos CLDATA formátumban a követelményjegyzékben meghatározott utasítások, így pl.: a fordulatszám, előtolás, szerszámcseré, előkészítő funkciók és koordinátaértékek. Az algoritmus elkészítése során különös figyelmet fordítottunk arra, hogy az adott programban szereplő, ismétlődő utasítások mindegyike visszanyerhető legyen, valamint arra is nagy hangsúlyt fektettünk, hogy a kapott CLDATA adatok a megfelelő formátumban álljanak rendelkezésre az inverz posztprocesszáshoz.



3. ábra: Az inverz posztprocesszor átfogó algoritmus

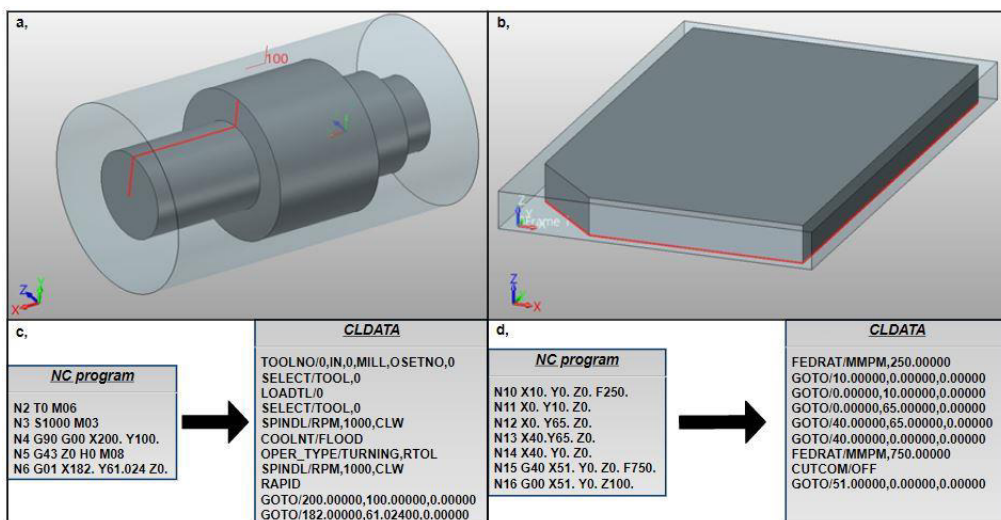
Az NC programon belül szereplő, különböző parancsok visszaalakításához, eltérő feltételek megadása szükséges, ezért az algoritmuson belül szükség van különböző sub rutinokat (alprogramokat) létrehozni. Ilyen rész-algoritmusra látható példa a 4. ábrán, ahol jól megfigyelhető, hogy az egyes feltételek teljesülése esetén a keresett NC mondatra válaszul az azonos információt tartalmazó CLDATA adatot kapjuk vissza. A rész-algoritmusok esetében a legnagyobb kihívást az jelentette, hogy míg az NC mondatok adott soron belül több utasítást is tartalmaznak, addig a CLDATA nyelvben csak egynemű adatok fordulnak elő. Minden esetben fontos volt tehát, hogy az ilyen esetekben az utasítást, a CLDATA nyelvnek megfelelően, logikusan, egymás után adja vissza az inverz posztprocesszor.



4. ábra: Példa a fordulatszám algoritmusának működésére adott NC mondat (fehér út) esetében

4 Validálás

Az inverz posztprocesszor validálása során azt tartottuk szem előtt, hogy a folyamat segítségével kapott CLDATA a későbbiekben szabadon illeszthető legyen bármely szerszámgép vezérléséhez. Ennek érdekében a kapott CLDATA adatokat az eredetivel összevetve más NC gépek vezérlőjéhez posztprocesszáltuk, ezáltal bemutatva, hogy az inverz posztprocesszor alkalmazásával sikeresen lehet egy adott vezérlő környezetéhez illesztett adatokat egy másik vezérlő környezetéhez illeszteni. Az 5. ábrán látható alkatrészekben a ZW3D CAD/CAM szoftver segítségével egyszerű szerszámmozgásokat programoztunk, majd ezeket a kiválasztott Fanuc és Milltronics vezérlőkhöz posztprocesszáltuk, így megkapva az NC programot (5.c és 5.d ábrák). Ezt követően az általunk fejlesztett algoritmus segítségével, Visual Studio fejlesztőkörnyezetben C# nyelven létrehoztunk egy felhasználóbarát inverz posztprocesszort, amellyel a korábban megkapott NC programot sikeresen visszaalakítottuk CLDATA adatokká, ahogy ez az 5.c és 5.d ábrán is látható. A kapott CLDATA tehát ilyen módon később ismételt posztprocesszálható a kívánt vezérlőhöz a CAM szoftverek létező algoritmusaival.



5. ábra: Inverz posztprocesszor validálása

5 Összefoglalás

Kutatómunkánkban olyan inverz posztprocesszor fejlesztését kezdtük meg, mely algoritmussal egy NC program átalakítható szabványos CLDATA programmá. Az általunk fejlesztett inverz posztprocesszor segítségével —alapvető műveletelemek esetében— sikeresen tudtunk különböző NC programokat CLDATA nyelvre visszaalakítani, majd másik vezérlő környezetéhez illeszteni meglévő CAM posztprocesszorokkal. Az általunk fejlesztett inverz posztprocesszor lehetőséget ad arra, hogy megvalósítsa a CLDATA és CNC program közötti kétirányú kommunikációt, ezáltal növelve a CAD/CAM/CNC munkafolyamatok gyorsaságát, hatékonyságát és rugalmasságát.

6 Hivatkozások

- [1] Jozef Majerik, Jaroslav Jambor (2015). Computer Aided Design and Manufacturing Evaluation of Milling Cutter when High Speed Machining of Hardened Steels. *Procedia Engineering*, 100 (2015), 450-459. <https://doi.org/10.1016/j.proeng.2015.01.390>
- [2] Boór Ferenc, Hervay Péter, Kakuk Gyula, Markos Sándor, Mátyási Gyula, Mikó Balázs (2012). CAM Tankönyv. Typotex kiadó
- [3] Johannes Schmid, Rudolf Pichler (2020). Seamless Data Integration in the CAM-NC Process Chain in a Learning Factory. *Procedia Manufacturing*, 45 (2020), 31-36. <https://doi.org/10.1016/j.promfg.2020.04.038>
- [4] Carsten Schaede, Stefan Seifermann, Joacim Metternich (2018). Automated generation of CNC programs for manufacturing of individualized products. *Procedia CIRP*, 72 (2018), 1251-1257. <https://doi.org/10.1016/j.procir.2018.03.064>