

IPARI SZIMULÁCIÓS KUTATÁSOK A MISKOLCI EGYETEM ALKALMAZOTT INFORMATIKAI TANSZÉKÉN

RESEARCHES BASED ON INDUSTRIAL SIMULATION AT THE DEPARTMENT OF INFORMATION ENGINEERING OF THE UNIVERSITY OF MISKOLC

*Dudás László, Hornyák Olivér, Kulcsár Gyula, Nehéz Károly**

ABSTRACT

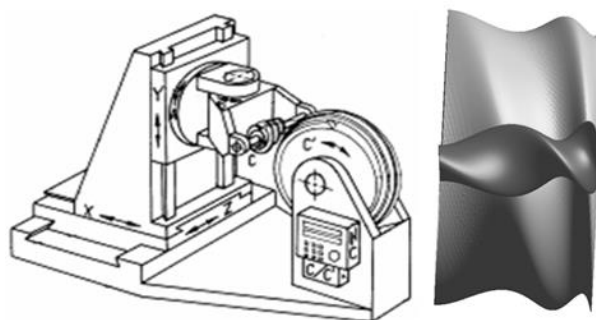
This paper describes the work has been done in various areas of simulation at the Department of Information Engineering of the University of Miskolc.

1. BEVEZETÉS

A számítógépre alapozott szimuláció napjaink igen széles körben alkalmazott kutatási módszere. A következőkben a Miskolci Egyetem Alkalmazott Informatikai Tanszékén folyó kutatások eredményeiről adunk egy rövid áttekintést.

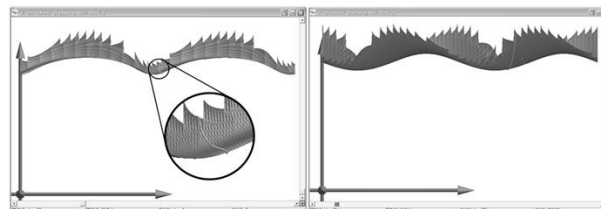
2. BONYOLULT FELÜLETEK ALÁMETSZÉSMENTES MEGMUNKÁLÁSA

A bonyolult felületek, elsősorban fogazatok működő felületeinek és szerszámfelületeinek tervezésére kidolgozott Surface Constructor kinematikai modellező és szimulációs szoftver egyik új alkalmazása során speciális forgódugattyúval és forgókamrával rendelkező kompresszor, illetve expanziós gép forgódugattyújának köszörüléssel történő befejező megmunkálását modelleztük. A technológia az 1. ábrán látható, szokásostól eltérő nem forgásfelület alakú korongot, speciális köszörűgépet és technológiát alkalmaz. A feladat a köszörűkorong felületének és a szerszámbeállítás geometriai paramétereinek meghatározása volt. A korong felületét a munkadarabbal való burkolással állítjuk elő. Mivel a megmunkálás közben nem megfelelő paraméterek mellett könnyen alámetszés léphet fel a korong és a munkadarab felülete között, olyan beállítást kellett megtalálni, mely biztosítja az alámetszés-mentességet. Első megközelítésben ezt a korong és a munkadarab között minden pillanatban fennálló felület-felület érintkezés, azaz folytonos érintkezési görbe vizuális ellenőrzésével próbáltuk elérni, de ennek kivitelezése csak közelítőleg,



1. ábra Speciális köszörűgép és a speciális korong által megmunkált kompresszor rotor

diszkrét szöghelyzetek sokaságánál sikerült. $\Gamma = 3^\circ$ korongbedöntési szög megfelelőnek tűnt. A Surface Constructor egyik speciális nézete, az $R=R(\Phi)$ függvények megjelenítése pontosabb ellenőrzésre adott lehetőséget. Ebben a nézetben a korong felületének minden pontját egy $R=R(\Phi)$ görbe jellemzi. A hibátlan felület-felület kapcsolódáshoz az szükséges, hogy ez a görbe egy sima minimumponttal rendelkező görbe legyen. A korong egy felületi görbéjének pontjait jellemző $R=R(\Phi)$ görbéknek pedig egy völgy alakzatot kell adniuk. Amint a 2. ábra bal oldalán látható, $\gamma = 3^\circ$ fok esetén ez nem minden pontban teljesült. Változtatva a γ korongbedöntési szöget, meglepő eredményként $\gamma = 0^\circ$ fok esetén adódott a megfelelő alámetszésmentes korongalak. [1]



2. ábra A speciális $R=R(\Phi)$ függvényeken az inflexió alámetszést jelez. Az alámetszésmentességet egy völgyfelület jelzi a korong egy felületi görbéjére

* egyetemi docens, Miskolci Egyetem, Alkalmazott Informatikai Tanszék

3. FORGÁCSOLÓ ERŐ SZIMULÁCIÓJA NEURÁLIS HÁLÓVAL

A forgácsolási műveletek szimulációjánál alkalmazható modellek lehetséges 6 absztrakciós szintje a következő:

1. fizikai (physical) szint,
2. műveletelem (element) szint,
3. alaksajátosság (feature) szint,
4. művelet (operation) szint,
5. munka (job) szint,
6. rendelés (order) szint.

Az első három modellezési szinten a technológia fizikai folyamatai, a folytonos állapotjelzők játsszák a fő szerepet. Itt tehát olyan modellekre van szükség, amely az anyagválasztás geometriai, kinematikai, dinamikai, fizikai tulajdonságait prezentálja. (Nem foglalkozunk itt most a további három szint modelljeivel, amelyek a eseményvezérelt diszkrét dinamikus modellezés (DES) hatáskörébe tartoznak.)

A forgácsolás elmélet, a technológiai tervezés elmélete, valamint a számítógépes NC programozás által kidolgozott eljárások alkalmasak a modellezési feladatok egy részének megoldására. Ezek geometriai és fizikai modellek. A hagyományos megoldások azonban nem támogatják a forgácsolási műveletek műszaki-gazdasági modellezését és szimulációját. A felmerülő nehézségek okai a következők:

- A modelleknek nagyszámú bemeneti, kimenő és belső változót kell kezelni.
- A változók közötti kapcsolatokra számos esetben nincs megbízható elméleti eredmény.
- Az anyagtulajdonságok figyelembevétele igen nehéz, csak empiriával lehetséges.
- A részfolyamatok között néhány csak sztochasztikus változókkal írható le.

Az esztergálási műveletelemek részletes folyamat modelljére számos korábbi technológiai és NC programozó rendszerben készültek modellek.

Az Alkalmazott Informatikai Tanszéken kidolgozott modell 3 lényeges tulajdonságában tér el a korábbiaktól. Ezek a következők:

- Az anyagválasztás geometriájának modellezésében a térfogati modell és a forgácsolási intenzitás alkalmazásának elvét követi, és a megtervezett NC program pontos szerszám-pálya adataira támaszkodik.
- A technológiai jellemzők tekintetében a nemlineáris modellrészeknél AI módszereket, főként neurális hálót használnak.
- A műveletelemek aggregálható (integrálható) indexeit, amelyek kulcsfontosságúak a műveletek gazdasági-menedzseri értékeléséhez, numerikus integrálási módszerekkel, viszonylag pontosan állítja elő.

3.1 Folyamatmodellezés neurális hálókval

Az esztergálási műveletelem nemlineáris folyamatainak modellezésére célszerű hibrid, többszintű neurális háló modellt bevezetni. Hibrid modellről beszélhetünk, mivel bizonyos paramétereket neurális hálóval becsülünk (például a forgácsoló erőt), más paramétereket pedig a becsült értékek felhasználásával analitikus képletek segítségével számolunk. Ilyen például a forgácsoló nyomaték számítása.

Az első szinten az alapvető geometriai és technológia állapotváltozók kerülnek kiértékelésre. Ezeket a paramétereket a szimulátor számítja a gépgyártás-technológiában jól ismert matematikai összefüggések alapján.

A második szinten az összetett determinisztikus állapotváltozók aktuális értékei kerülnek meghatározásra. Ezzel a geometriai folyamatmodellezés véget ér. A harmadik szinten a forgácsoló erők aktuális értékének becslése a legfontosabb feladat. Ezt a feladatot neurális háló valamint empirikus összefüggés alapján több módszerrel is számíthatja a szimulátor modellje.

Az erők ismeretében a negyedik szinten a forgácsoló nyomaték, a forgácsoló teljesítmény és a felhasznált energia számítása lehetséges. Ez utóbbi már integrált jellemző, mint ahogyan ilyen a műveleti idő is. Számítható a halmozott szerszámkopás, a még hátralévő éltartam, az átlagos és a maximális főforgácsoló erő, stb.

Az utolsó, ötödik szinten a menedzser indexek aktuális értékei és a technológiai jellemzők aggregált vagy átlagos értékeit számítjuk. A szimuláció végén ezek az esztergálási művelet minden gyártásirányítás szempontjából fontos tulajdonságát tartalmazzák.

A modelleket és algoritmusokat egy NC szimulátor alkalmazásban valósítottuk meg. [2]

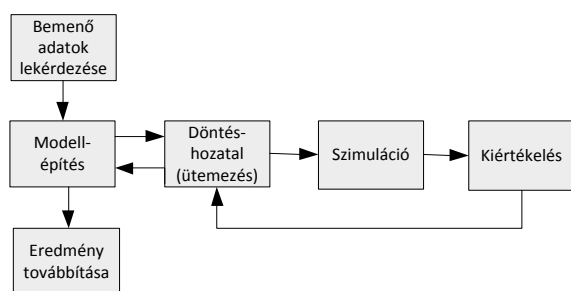
4. SZIMULÁCIÓRA ALAPOZOTT TERMELÉSPROGRAMOZÁS

A diszkrét termelési folyamatok műhelyszintű előidejű (prediktív) és valós idejű (reaktív) ütemezése mind elméleti, mind gyakorlati szempontból az erősen modellfüggő és komplex kombinatorikus optimalizálási feladatok közé tartozik. Ennek következtében nagyon sokféle ütemezési feladattípus ismert [3]. A napjaink ipari gyakorlatában egyre fontosabbá váló, rugalmas és igény szerinti gyártási folyamatok irányítása szükségessé teszi az ismert ütemezési modellek további jelentős kiterjesztését.

A termelési folyamatok finomprogramozásakor a termelő rendszer lehetőségeit, képességeit és korlátait figyelembe véve, a belső rendelések teljesítéséhez szükséges munkák időbeli végrehajtását kell megtervezni. Ennek során a szükséges erőforrások

allokálását és a feladatok végrehajtásának indítási időpontját kell megtervezni úgy, hogy a vállalat magasabb szintjén megfogalmazott célok és a gyártásirányítás járulékos saját belső céljai egyaránt megvalósuljanak.

A különböző ütemezési feladat-osztályokba besorolható feladatok megoldására egy többcélú, integrált, heurisztikus keresési technikára és gyors szimulációra alapozott megoldási módszert fejlesztettünk ki [4]. Munkánk során újabb és újabb gyártási jellemzőket, alternatívákat és korlátozásokat magukba foglaló termelésprogramozási modelleket dolgoztunk ki. A megoldási koncepció elvi vázlatát a 3. ábrán látható.



3. ábra Szimulációra alapozott termelésprogramozás

A bemenő adatok felhasználásával egy modellépítő komponens definiálja a rendszerben lévő objektumokat. Kiemelt fontosságú feladata a belső rendelésekhez kapcsolódó munkák és az érvényben lévő korlátozások definiálása. A szükséges rendelkezésre állási, alkalmazhatósági és megvalósíthatósági vizsgálatok elvégzését követően a modellépítő komponens felépíti a modellobjektumok között fennálló teljes kapcsolatrendszerét.

A belső rendelések ütemezési alapegységekre bontásával önálló munkák jönnek létre. Az ütemező modul minden egyes munkához hozzárendel egy megfelelő végrehajtási útvonalat, továbbá hozzárendel egy megfelelő gépet a kiválasztott útvonal minden egyes végrehajtási lépésének megfelelő gépcsoportból, és meghatározza minden érintett gépen a munkák végrehajtási sorban elfoglalt pozícióját. Ezáltal az ütemező algoritmus a munkákhoz hozzárendeli a konkrét feladatlistát, és így a gépeken a gyártási sorozatnagyságok és az azokat elválasztó átállítási műveletek a döntési változók függvényében ütemezés közben alakulnak ki. A gyártási modelltípustól függően bizonyos esetekben az ütemező az erőforrások rendelkezésre állási időintervallumait is döntési változóként kezeli, míg más esetekben ezek korlátozó feltételek formájában jelennek meg. Az ütemezési (döntéshozatali) folyamat eredményeként elkészül egy termelési ütemterv.

Az ütemtervben szereplő feladatok végrehajtásához kapcsolódó időadatok számítását egy megfelelően gyors

végrehajtás-vezérelt szimulációs algoritmus végzi el. A szimuláció figyelembe veszi az egyes gépek rendelkezésre állási időintervallumait, az egyes gépeken az adott munkák sorrendje által meghatározott átállítási időket, a munka-gép összerendelések alapján számítható megmunkálási időket és az egyéb kapcsolódó járulékos (pl. anyagmozgatási) időket. A szimuláció közben ismertté válik az egyes feladatok gépenkénti tervezett – és következményként a munkák, valamint a megrendelések származtatott – indítási és befejezési időpontja.

A szimuláció által számított időadatok és egyéb értékek felhasználásával egy értékelő komponens kiszámítja a megoldásra vonatkozó célfüggvény értékeit és teljesítménymutatókat. Ezek figyelembevételével az aktuálisan vizsgált megoldás és az addig legjobbnak ítélt megoldás egymáshoz viszonyított (relatív) minőségének számszerűsítésével kiértékelésre kerül a megoldás.

Az ütemező modul iteratívan módosítja az aktuális ütemtervet, konzisztens változtatásokkal új megoldásváltozatokat készít, majd szimulációt és kiértékelést követően a célfüggvény-értékektől függően tovább folytatódhat a legjobb megoldás keresése.

5. ANYAGLEVÁLASZTÁS SZIMULÁCIÓ IMPLICIT FELÜLETEKKEL

A CPU-t és a GPU-t más-más céllal fejlesztik, ezért eltérő feladatokat képesek hatékonyan végrehajtani. A CPU a számítógép központi egységként a folyamatok és szálak kezelését végzi hatékonyan. A szálak közötti gyors kontextus váltásokra és az egyes szálak szekvenciális végrehajtására tervezték. Ezzel szemben a GPU tervezésekor egyetlen folyamat szálainak párhuzamos végrehajtása a legfontosabb követelmény. Emiatt jól lehet alkalmazni olyan esetekben, ahol nagy adathalmazon kell azonos műveleteket végrehajtani és az egyes adatok között minimális a függőség. A GPU-k programozása szinte tetszőleges programozási nyelven történhet, csak a megfelelő függvénykönyvtárakat kell használni. Több ilyen függvénykönyvtár használata is elterjedt. A Compute Unified Device Architecture (CUDA), az nVidia párhuzamos programozási eszköztára, melynek segítségével csak az nVidia által gyártott grafikus kártyák programozhatóak. Ez az eszköztár könnyen használható GPUra optimalizált matematikai függvénykönyvtárakat tartalmaz C és C++ programozási nyelvhez.

Az Open Computing Language (OpenCL), egy olyan szabvány, melynek segítségével párhuzamosan programozható heterogén rendszerek. A heterogenitás ebben az esetben azt jelenti, hogy ugyanaz a kód képes módosítás nélkül futni bármelyik gyártó GPU-ján vagy CPU-n. Hasonló használhatóságot biztosít, mint a CUDA eszköztár, de annak megkötései nélkül.

A megmunkálás szimuláció problémájának matematikai leírása a következő [5]. Legyen M vonalszakaszok halmaza a térben, amely leírja a szerszámmozgást. Legyen f függvény a szerszám és alakjuk szerint ujjmaró (1) vagy gömbvégű maró (2).

$$f(x, y) := -r,$$

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

$$f(x, y) := -\sqrt{r^2 - x^2 - y^2}$$

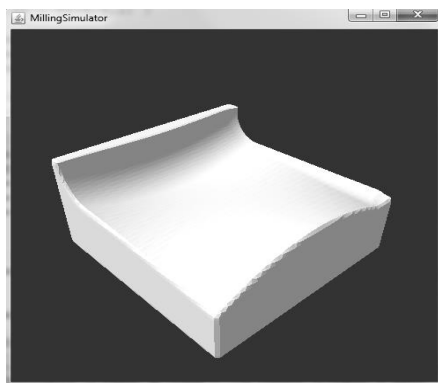
$$x, y \in D(f) := \{(x, y) \mid x^2 + y^2 \leq r^2\} \quad (2)$$

Mindkét esetben az origót tekinthetjük a programozott pontnak. Az x és y értékei leírják a szerszámmozgást a nyersdarab vízszintes síkjában, z pedig a magasságot jelenti. Ha a munkadarabot egy téglatestnek tekintjük x_{\min} , x_{\max} , y_{\min} , y_{\max} , z_{\min} , z_{\max} méreteivel megadva, akkor az anyagleválasztás matematikailag a következő implicit formulával írható le (3).

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

Az $F(x, y)$ függvény kiszámítása időigényes feladat lehet, főleg szabadformájú felületek esetén ahol a szerszámhelyzetek száma több tízezer is lehet. Az irodalomban egzakt és analitikus megközelítések léteznek $F(x, y)$ meghatározására. Jelen cikkben a OpenCL technológiát alkalmazunk GPU segítségével.

Az irodalomban implicit függvények megjelenítésére számos módszer ismert: az u.n. masírozó kockák algoritmus, valamint duális kontúrozás. A prototípusban a masírozó kockák algoritmusát alkalmaztuk, amelynek hátrány az élek automatikus letörése. A szimuláció lépései a következők: 1.) szerszám, munkadarab definíció implicit felületekkel;



4. ábra Gömbvégű szerszám pályaszimulációja ~1600 vonalszakasszal 2500ms számítási idővel (azaz: 1.5 ms szerszámhelyzetenként)

2.) szerszámhelyzetek előállítás; 3.) a (3) egyenlet kiszámítása az egyes szerszámhelyzetekre OpenCL segítségével, párhuzamosan;

A 4. ábrán a marási folyamat megjelenítését látjuk. Összehasonlítva a GPU alapú megoldással körülbelül 20 szoros sebességnövekedést tudunk elérni. Szimuláció környezet mindössze egy laptop, AMD Radeon HD 5700-as videokártyával, ami 4 GPU maggal rendelkezik. Az implementáció Aparapi OpenCL technológiát alkalmaz [6].

6. ÖSSZEZGÉS

A cikk négy számítógépes modellezéssel és szimulációval megoldott feladatot ismertetett. A részletek iránt érdeklődők figyelmét felhívjuk a hivatkozott irodalmakra is.

7. KÖSZÖNETNYILVÁNÍTÁS

A bemutatott kutató munka a TÁMOP-4.2.1.B-10/2/KONV-2010-0001 jelű projekt részeként az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg

8. IRODALOM

- [1] DUDÁS L.: Algorithm for Visualization of Undercut Situations of Surfaces Generated by Enveloping, Proceedings of the Iadis International Conference Applied Computing 2012 Madrid, Spain, October 19-21, 2012. pp. 291-298. ISBN: 978-989-8533-14-2
- [2] HORNYÁK O.: Esztergálási műveletek kiterjesztett számítógépes szimulációja intelligens módszerek alkalmazásával. PhD disszertáció 2002.
- [3] LEI D. (2009) Multi-objective production scheduling: a survey. The International Journal of Advanced Manufacturing Technology, 43 (9-10):926-938.
- [4] KULCSÁR GY., KULCSÁRNÉ F. M. (2012) Detailed Scheduling Based on Simulation. XXVI microCAD International Scientific Conference, Miskolc, Hungary, p. 6. (CD).
- [5] H. MÜLLER, F. ALBARSMANN, A. Zabel: Efficient raster-based simulation and visualization of 3-axis Milling of Free-form Shapes. Research report No. 667/1998, Fachbereich Informatik Universität Dortmund, Germany, 1998
- [6] APARAPI, API for data parallel Java. Allows suitable code to be executed on GPU via OpenCL, <http://code.google.com/p/aparapi/>, 2012.