Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Robotos pakolási műveletek hatékony tervezése

Szakdolgozat

Készítette
Bodor Máté

Konzulens
dr. Kovács András
dr. Hullám Gábor

2019. november 18.

Tartalomjegyzék

HALLGATÓI NYILATKOZAT

Alulírott *Bodor Máté*, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2019. november 18.

Bodor Máté
hallgató

Kivonat dec 6

Jelen dokumentum egy diplomaterv sablon, amely formai keretet ad a BME Villamosmérnöki és Informatikai Karán végző hallgatók által elkészítendő szakdolgozatnak és diplomatervnek. A sablon használata opcionális. Ez a sablon LaTeX alapú, a *TeXLive* TeX-implementációval és a PDF-LaTeX fordítóval működőképes.

Abstract

This document is a LaTeX-based skeleton for BSc/MSc theses of students at the Electrical Engineering and Informatics Faculty, Budapest University of Technology and Economics. The usage of this skeleton is optional. It has been tested with the *TeXLive* TeX implementation, and it requires the PDF-LaTeX compiler.

1. fejezet

Irodalmi áttekintés nov 19

1.1. Robotok kinematikai modellezése

1.1.1. Feladat tér

A robotikában a feladat tér egy pontját egy M 4x4-es homogén transzformációs mátrixszal adjuk meg. Ez a mátrix a robotkar rögzítési pontjához kötött koordináta rendszer és a robotkar végpontjához kötött koordináta rendszer között add meg egy transzformációt.

$$M = \begin{bmatrix} R_{3x3} & T_{3x1} \\ 0_{1x3} & 1 \end{bmatrix}$$

Az M mátrix egy 3x1-es T részmátrixa írja le a rögzítési ponthoz kötött Descartes koordináta rendszerben, hogy hol található a végpont koordinátarendszere. Azt, hogy a végponthoz viszonyított koordinátarendszer, hogy helyezkedik el a rögzítési ponthoz képest, azaz a két koordinátarendszer megfelelő tengelyei milyen szöget zárnak be egymással,egy 3x3-as R részmátrix írja le. Ezzel az R részmátrixal lényegében a robotkar végének állását írjuk le.

[Robotic Task Sequencing Problem: A Survey]

1.1.2. Robotkonfigurációs tér

A robotkar állását, konfigurációját megadhatjuk a robotkar egyes csuklóinak állásával. Ekkor C legyen az egyes csuklók állásának szögértéke. Egy 3 szabadságfokú robotkar esetén C három darab értéket fog tartalmazni.

A robotkarkonfigurációs teret szokás még csukló térnek vagy robot térnek nevezni.

[KÉP]

[Robotic Task Sequencing Problem: A Survey]

1.1.3. Denavit–Hartenberg paraméter

A robotkar geometriai reprezentációját többféleképpen is megtudjuk adni. Az egyik kényelmes módja, hogy minden egyes kartaghoz egy koordinátarendszert rögzítünk. Jacques Denavit és Richard Hartenberg írta le először egy konvencionális jelölésrendszert, hogy általánosítsa a kartagokhoz rendelt koordinátarendszerek leírását.A jelölésrendszer elemeit az 1955-ös publikációjuk alapján Danavit-Hartenberg paramétereknek (D-H paramétereknek) nevezzük őket.

Ebben a jelölésrendszerben négy paraméterre van szükségünk, hogy a egy koordinátarendszert leírjunk egy másik koordináta rendszerhez viszonyítva. A négy paraméterből kettő vonatkozik a csuklókra és kettő a kartagok leírására. A csuklóknál a hozzájuk kapcsolódó kartagok eltolása d és a csuklóállás szöge Θ . A kartagoknál a kartag hossza a és a kartag elcsavarodása α .

A jelölésrendszebren az i-edik csukló az i-edik kartag végén helyezkedik el, az i-edik és i+1-edik kartag között. A csukló eltolás d_i és a csuklószög Θ_i az i-1.csukló szerint van mérve, ezért a csukló indexe és a csukló paramétereinek indexe nem egyezik meg.

[KÉP]

[Hivatkozás: Springer Handbook of Robotics 2.6 WIkipedia: $https : //en.wikipedia.org/wiki/Denavit - Hartenberg_parameters$]

1.1.4. Forward kinematika

A forward kinematika feladata nyílt kinematikai láncú robotkar esetén, hogy a kar csuklóállásainak értékéből, egy viszonyítási ponthoz tekintve, meghatározza a robotkar végpontjának helyzetét. A viszonyítási pont általában a robotkar rögzítési pontja, ami egy konstans eltolással megkapható a 0. kartagból. Általában a robotkar végére valamilyen eszközt szerelünk, hogy befolyásolni, vagy érzékelni tudja a környezetét. A robotkar végpontjának, tool frame-jének, a felszerelt eszköz végpontját szokás hívni. A tool frame megkapható egy konstans transzformációval a robot utolsó kartagjához képest.

A számításhoz bemeneti paraméterként két dologra van szükség, egyik a robotkar fizikai paraméterei, a másik a robotkar csuklóállásainak értéke.

A forward kinematikai probléma megoldása egy transzformáció egy a robotkar rögzítési pontjához kötött koordináta rendszerből a tool-framehez kötött koordináta rendszerbe, azaz a felszerelt eszköz és a rögzítési pont között. Ezt a transzformációt egy 4x4-es homogén transzformációs mátrixszal szokás leírni. A transzformáció egyértelmű, mert egy robotkarkonfigurációhoz egy homogén transzformációs mátrixot rendel.

[Hivatkozás: Springer Handbook of Robotics 2.6]

1.1.5. Inverz kinematika

1.2. Sorrendtervezés

1.2.1. GTSP

1.2.2. LNS

1.2.3. Megoldó algoritmusok

1.2.3.1. Lokális keresés

Az informatikában a lokális keresés egy heurisztikus módszer nagy számításigényű optimalizációs feladatok megoldására. A lokális keresés segítségével megtalálhatjuk a legjobb megoldást egy probléma megoldásainak halmazából. Egy lokális kereső algoritmus a megoldási térben, megoldásról megoldásra halad, lokális változásokat végrehajtva. Az algoritmusok addig futnak, amíg el nem érnek egy olyan megoldáshoz, amin már nem tudnak tovább javítani vagy amíg a keresés ideje le nem telik.

Lokális keresést nagyon sok féle nagy számítási igényű feladaton alkalmaznak, beleértve az utazó ügynök problémát is. Optimalizációs problémák megoldása során a lokális keresés egy célfüggvény segítségével keresi meg a megoldások közül a legjobbat. Ha az állapotteret és a célfüggvényt egy diagramon ábrázoljuk, akkor a feladat állapottérfelszínét kapjuk. Egy teljes lokális keresés mindig talál megoldást, ha az létezik. Az optimális lokális keresési algoritmusok mindig megtalálják a keresett globális minimumot vagy maximumot.

1.2.3.2. Hegymászó keresés

A hegymászó keresés egy egyszerű mohó algoritmus, ami mindig az aktuális legjobb érték felé lép. Ha az algoritmus a következő lépésben már nem tud javítani, akkor a keresés megáll. A hegymászó keresés nem optimális lokális keresés. Mivel a keresés nem járja be a teljes állapotteret, csak amíg az eredményen javítani tud, a keresés könnyen lokális optimumban ragadhat.

1.2.3.3. Szimulált lehűtés

A szimulált lehűtés a hegymászó algoritmus továbbfejlesztett változata. A hegymászó algoritmussal a legnagyobb probléma, hogy könnyen egy lokális optimumba juthatunk és befejeződik a keresés. Ez abból adódik, hogy a keresés során az állapottérnek csak minimális részét járjuk be. A szimulált lehűtés ezt próbálja meg kiküszöbölni. A hegymászó algoritmus mindig a legjobb lépést választja, ehelyett a szimulált lehűtés egy véletlen lépést választ. Ha a lépés javítja a célfüggvény értékét, akkor mindig végrehajtásra kerül. Ellenkező esetben az algoritmus a lépést csak P eséllyel teszi meg. A P valamilyen 1-nél kisebb szám, értéke exponenciálisan csökken a lépés értékének rosszaságával, azaz egy delta E mennyiséggel, amivel a célfüggvény értéke romlott. A valószínűség a szimulált lehűtés másik T paraméterétől is függ. A T hőmérséklet csökkentésével is csökken a rossz lépés valószínűsége. A rossz lépések esélye a szimulált lehűtés indulásánál, amikor a T nagy, elég magas, a T csökkenésével viszont egyre valószínűtlenebbekké válnak. Ha a T értékét kellően lassan csökkentjük akkor matematikailag bebizonyítható, hogy a szimulált lehűtés közel egy valószínűséggel megtalálja a keresett globális optimumot.

Az algoritmus a 80-as évek elején terjedt el. A szimulált lehűtést először a VLSI nyomtatott áramkörök tervezésekor használták. Azóta széleskörben alkalmazzák a nagy számításigényű optimalizációs feladatokra.

1.2.3.4. Tabu keresés

A tabu keresés lényegében egy rövidtávú memóriával rendelkező hegymászó keresés, pár szabállyal kiegészítve. A tabu keresés során addig folytatjuk a keresést amíg ki nem elégítjük a keresés feltételét, ez lehet egy célfüggvényérték vagy a futási idő korlátozása. A tabu keresési algoritmus, ha lehet mindig olyan lépést választ, amivel javít a célfüggvény értékén. Ha nincs ilyen lehetőség akkor rontó lépést is elfogad. Az algoritmus egy véges nagyságú FIFO memóriában tárolja el a már meglátogatott csomópontokat. Ha egy csomópont benne van a memóriájában akkor oda nem lép az algoritmus. Az algoritmus képes arra, hogy kitörjön lokális optimumból. Ennek hatékonysága a FIFO méretétől függ, addig elkerüli az algoritmus a lokális optimumot amíg a csomópont a memóriában van.

A tabu keresés algoritmust a 1989 években írta le Fred W. Glover. A keresési eljárást azóta széleskörben alkalmazták a logisztika, orvosbiológiai elemzés, ütemezés és egyéb nagy számítási teljesítményű optimalizációs feladatnál.

1.3. Pályatervezés

2. fejezet

Feladat definíció nov 15

A szakdogalzati feladatom egy közel valós idejű sorrend és pályatervezési általános megoldó készítése, robotos pakolási feladatokhoz. A pakolási feladatot egy darab nyílt kinematikai láncú robotkar hajtja végre. A robotkar munkatere előre adott. A pakolási feladat munkadarabjai egy sík felületű tálcán helyezkednek el. Ez a tálca egy rázóasztalra van erősítve, ennek segítségével a munkadarabokat jól el lehet különíteni, az esetek nagy részében így nem fedik egymást, a robotkar hozzájuk tud férni. A pakolási feladatott előre adott számú munkarabbal hajtjuk végre. A munkadarabok célhelyzete előre meghatározott. A pakolási feladat megoldása során tetszőleges sorrendben fel kell vennünk az egyes munkadarabokat és a célhelyzetükbe kell vinnünk őket. A pakolási feladatnak vége, ha az összes munkadarabot eljuttatjuk a célhelyzetébe. Ekkor a tálcán nem találhatóak munkadarabok. Ha a tálcára új munkadarabokat helyezünk, akkor azoknak a pakolását egy következő feladatnak tekintjük.

A megoldó a szükséges bemeneti paraméterek magadása után, ilyen például a munkadarabok helyzete, egy a robotkar által ütközésmentesen bejárható pályát ad vissza. Ez az út időben egy közel optimális megoldása a munkadarabok célba juttatásának.

Mivel közel valós időben szeretnénk végrehajtani a pakolási feladatot, ezért célunk egy olyan megoldó tervezése, hogy a megoldó futásideje és a talált pálya bejárási idejének összege a lehető legkisebb legyen.

A feladat megoldás három részre bontható, ez a három:

- inverz kinematika,
- sorrendtervezés és robotkarkonfigurációs lista választás,
- pályatervezés.

[KÉP]

2.1. Bemeneti paraméterek

A feladat megoldása során szükségesek bizonyos bemeneti paraméterek, amiket a megoldás során állandónak tekintünk. Ezek az értékek lehetnek előre adottak vagy az érzékelők adatai alapján generáltak. Az adott, állandó konstans érték, lehet például a környezet modellje vagy a robotkar egyes paraméterei.

2.1.1. Munkadarabok kiinduló- és célhelyzete

A munkadarabok kiindulási helyzete egyes érzékelők adatai alapján meghatározhatók. Az egyik leghatékonyabb módszer kamera segítségével meghatározni az egyes munkadarabok helyzetét.

Nem elegendő egy munkadarab pontos helyének leírásához egy a robotkar rögzítési pontját origónak tekintő Descartes koordináta rendszert használni. Mivel a munkadarabok nem pontszerű testek, így nem elég tudni csak a munkadarabok tengelyekhez mért távolságát. Egy munkadarab több félé pozícióban helyezkedhet el aszerint, hogy hány stabil egyensúlyi helyzete van.

A munkadarabok pontos helyének leírásához az elméleti összefoglalóban már említett világkoordináta rendszert használjuk. Ekkor a térben az egyes munkadarabok helyzetét egy két dimenziós homogén transzformációs mátrix írja le. Az egyes munkadarabok kiinduló helyéhez egy ilyen mátrixot rendelhetünk bemeneti paraméterként. Legyen ennek a kiindulóhely mátrixnak az i-edik munkadarabhoz rendelt jele WS_i.

A munkadarabok célhelyzete előre meghatározott. Megoldandó feladattól függ, hogy az egyes munkadarabokhoz ugyanazt vagy egyedi célhelyzetet rendeljünk. Az általánosabb használhatóság miatt, minden egyes munkadarabhoz célszerű egyedi értéket rendelnünk. A kiindulóhelyzethez hasonlóan a célhelyzetet is világkoordináta rendszerben adjuk meg. Legyen az i-edik munkadarab célhelyzet mátrixának jele WE_i.

2.1.2. Robotkar kiinduló- és végállapota

A hagyományos nyílt hurkú ipari robotok mozgásukat előre beprogramozott ciklus szerint végzik így a kiinduló- és végállapotuk megegyezik. Esetünkben ez nem mindig van így, ezért a robotkar mozgásának tervezésekor szükségünk van a robotkar kezdeti és végállapotára.

A robotkar egy állását az egyes csuklók szögeinek megadásával jellemezhetjük. Legyen egy n csuklóból álló robotkar egyes csuklóinak szöge r_1, r_2, r_3, ..., r_n. Ekkor a csuklók szögei egyértelműen meghatároznak egy R robotkar konfigurációt, azaz $\mathrm{R} = [\mathrm{r}_1, \mathrm{r}_2, \mathrm{r}_3, ..., \mathrm{r}_n]$. A robotkar kiinduló és végállapotát is egy ilyen R robotkar konfigurációval adhatjuk meg. Legyen a kiindulóállapotát leíró robotkar konfiguráció R_s és a végállapot leíró pedig R_e.

Abban az esetben ha R_s és R_e megegyezik a robotkar mozgása során pályáról, abban az esetben ha különböznek útról beszélhetünk.

2.1.3. Robotkar és a munkatér modellje

A robotkarmozgását leíró kinematikai számításoknál szükségünk van a robotkar modelljére. A robotkar térbeli leírását úgy kapjuk meg, hogy a robotkar, különálló, különmozogni képes részeit egy térbeli testtel közelítjük. Nem elég tudnunk ezeket a térbeli testeket. Szükségünk van még egy szabályrendszerre amely leírja a testek elhelyezkedését, mozgását és egymással való kapcsolatukat.

Egyes robotkaros problémák megoldása során szükségünk lehet a robotkar munkaterében lévő környezet leírására. Itt is a különálló részeket egy térbeli testtel közelítjük és egy szabályrendszerrel leírjuk az egyes testek elhelyezkedését és környezetükkel való kapcsolatát.

Esetünkben a robotkar, munkadarabok és munkaasztal leírására van szükség.

2.1.4. Robotkar általános paraméterei

Ahhoz hogy a robotkar mozgását vizsgáljuk szükséges tudnunk a mozgását befolyásoló paramétereket. Egy robotkar csuklókból és a csuklókat összekötő kartagokból áll. Az utolsó kartag végére jellemzően valamilyen eszközt rögzítenek, amivel a robot tudja érzékelni vagy befolyásolni környezetét. Robotkaros pakolási feladatoknál általában ez a munkadarabok megfogásához szükséges eszköz.

A tervezés és modellezés során a robotkar gyári adatait használjuk. Fontos azonban kiemelni, hogy a robotkarok rendelkeznek a gyártási pontatlanságokból adó eltérésekkel.

Ezen eltérések nagyban befolyásolhatják a robotkar mozgását. Ezért éles használat előtt korrigálni kell ezen adatokat.

A tervezés során a SZTAKI-ban megtalálható UR5-ös robotkart vettem alapul. De a feladat megoldása során törekedtem az általános felhasználhatóságra. Egy robotkart nagyon sok paraméterrel jellemezhetünk. A megoldás során azonban eltekintünk bizonyos paraméterektől, amik nem, vagy kis mértékben befolyásolják a robot mozgását. Ilyen például a robotkar terhelhetősége és az ebből származó mozgás változás.

2.1.4.1. A kar csuklóinak száma

A robotkar csuklóinak száma nagyban befolyásolja egy robot kar mozgási képességeit. Egy egy csuklóból álló robotkar egy tengely mentén képes mozgást végezni. A robotkar csuklószámának növelésével nő a robotkar mozgásának bonyolultsága, több tengely, dimenzió mentén képes mozogni a kar. Jelölje a robotkar csuklóinak számát NJ. UR5-ös robot esetében az NJ értéke 6.

2.1.4.2. A kar mozgását befolyásoló paraméterek

A robotkar mozgását a kar csuklóinak paramétereivel jellemezhetjük. A feladat megoldása során két ilyen csuklóparamétert használtam fel, ezek a paraméterek a csukló szöggyorsulása és szögsebessége.

Általánosságban a robotkar csuklója mozgása során először maximálisan gyorsul amíg el nem éri a maximális sebességét, ezt követően tartja ezt a sebességét, majd fékez és megáll. A fékezés értéke megegyezik a maximális gyorsulásával. Ezért egy csukló szöggyorsulását és szögsebességét a csukló maximális szöggyorsulásával és szögsebességével közelítettem.

Tapasztalatok szerint az UR5-ös robotnál ezek az értékek jó közelítést adnak. Ha valamiért a robotkar nem tudná tartani ezeket az értékeket, például túlterheljük a kart, akkor a robot mozgása leáll.

UR5-ös robotkar esetén ezek az értékek: [Táblázat]

2.1.4.3. Denavit-Hartenberg paraméterek

A Denavit-Hartenberg(D-H) paramétereket a robot kinematikai leírására használjuk, segítségükkel könnyen átválthatjuk a robotkar csuklóállásaival leírt pont helyét világkoordináta rendszerbe. A feladat során az inverz kinematikánál és a pályatervezésnél használom fel. UR5-ös robot esetén ezek a paraméterek: [Táblázat]

2.1.4.4. Megfogó paraméterei

A robotkar végén elhelyezkedő eszköz esetünkben egy a pakolási feladat munkadarabjait megfogni képes megfogó. A munkám során két paraméterét használta fel, egyik a megfogó eszköz hossza, jelölje Gh a másik a megfogó pofák egymástól való távolsága ez legyen Gw. Ezeket az adatokat a pályatervezésnél és az inverz kinematikánál használom fel.

2.2. Feladat definíció

A szakdolgozat során három részfeladatot kell megoldanom, ezeket a fejezet bevezetésében már említettem. Az egyes részfeladatok kimenetele lesz az azt követő részfeladat egyik bemenete. A szakdolgozatom során megtervezem és egymásba fűzöm ezeket a feladatrészeket. A következőkben szeretném definiálni a részfeladatokat.

2.2.1. Inverz kinematika

Az első részfeladat ami szükséges megoldani a munkadarabok kiinduló és célhelyzetének átszámítása robotkar által értelmezhető vonatkoztatási rendszerbe. Ekkor egy munkadarab világkoordináta rendszerbeli elhelyezkedését kell átváltanunk robotkonfigurációba. Egy ilyen transzformáció nem egyértelmű, egy világkoordinátához több robotkar konfiguráció is tartozhat.

Az inverz kinematikai számítás bemenete egy világkoordináta mátrix és a robotkar fizikai paraméteri. A kimeneti paramétere pedig egy lehetséges robotkar konfigurációkat tartalmazó lista. Ebben a listában azok a robotkar konfigurációk vannak benne, amiket egy ideális, kiterjedés nélküli robotkar képes felvenni, a képzetes gyökkel és nagy számítási hibával rendelkező konfigurációkat nem tartalmazza.

Az inverz kinematikai transzformáció azonban nem vizsgálja a környezet és robot kölcsönhatását. Ahhoz, hogy megkapjuk a lehetséges robotkar konfigurációkat, a listából ki kell szűrnünk azokat a konfigurációkat ahol a robotkar ütközik önmagával vagy a környezettel.

[KÉP]

2.2.2. Sorrendtervezés és robotkonfigárációs lista választás

A sorrendtervezés és a robotkonfiguráció választás két különálló feladat, de esetünkben ezek összekapcsolódnak. A részfeladat megoldása során törekszünk a időben legrövidebb sorrend kiválasztására. Mivel egy munkadarabhoz több robotkarkonfiguráció is tartozik, nem tudjuk munkadarabok sorrendjét a robotkarkonfigurációk figyelembe vétele nélkül meghatározni.

A részfeladat megoldása során egy döntést kell hoznunk, hogy melyik szabályos robotkarkonfigurációs sorrendet választjuk. A pakolási feladatot szeretnénk időben a legrövidebb idő alatt végrehajtani, ezért azt a sorrendtervezési és robotkarkonfigurációs lista választási eljárást keressük ami futási időben és megoldás idejét tekintve időben a lehető legkedvezőbb. Mivel nagyon sok jó megoldása létezik a feladat megoldásának és közel valós időben szeretnénk megoldani a pakolási feladatot, célszerű valamilyen heurisztikát használni a számunkra legkedvezőbb megoldás kiválasztásához. A részfeladat kimenetele a döntés eredménye lesz.

Legyen a kiválasztott robotkarkonfigurációs lista C és a lista i-edik eleme C_i.

[KÉP]

2.2.3. Pályatervezés

A pályatervezés során két robotkarkonfiguráció között szeretnénk egy időben optimális ütközésmentes pályát találni.

A pálytervezéshez bemenetként szükségünk van a robot és környezet modelljére és a robot paramétereire. A pályatervezést mindig ugyan annak a robotnak a munkaterében szeretnénk végrehajtani.

[KÉP]

Olyan megoldást kell keresni ami gyors, mivel közel valós időben szeretnénk végrehajtani a pakolási feladatot. Ezért nem feltétlenül az időben legoptimálisabb pályát keressük, hanem egy olyan eljárást aminek a véghajtási ideje és a talált pálya bejárási ideje együtt a lehető legkisebb. Esetünkben a pályatervezést a C lista minden egyes egymást követő elemére végre kell hajtanunk, ez $\mathrm{NJ} * 2 + 1$ darab tervezést jelent. A pályatervezési eljárás kimenete legyen egy lista amiben az egyes tervezések eredményit egymás után fűzzük.

Legyen ennek a listának neve P és a lista i-edik eleme P_i.

3. fejezet

Megoldás nov 26

3.1. Munkafolyamat

3.2. Inverz kinematika

3.3. Sorrendtervezés és konfiguráció választás

3.4. Pályatervezés

4. fejezet

Implementáció nov 22

ijm

5. fejezet

Eredmények dec 3

Eredmények

6. fejezet

Értékelés dec 3

Értékelés

Köszönetnyilvánítás

Ez nem kötelező, akár törölhető is. Ha a szerző szükségét érzi, itt lehet köszönetet nyilvánítani azoknak, akik hozzájárultak munkájukkal ahhoz, hogy a hallgató a szakdolgozatban vagy diplomamunkában leírt feladatokat sikeresen elvégezze. A konzulensnek való köszönetnyilvánítás sem kötelező, a konzulensnek hivatalosan is dolga, hogy a hallgatót konzultálja.

Függelék