

MISKOLCI EGYETEM
GÉPÉSZMÉRNÖKI ÉS INFORMATIKAI KAR



Automatizálási és Infokommunikációs Intézet

Villamosmérnöki (BSc) alapszak

**IPARI ROBOT TERVEZÉSE PICK AND PLACE
FELADATRA LABOR KÖRNYEZETBEN**

Szakdolgozat

BERTA ISTVÁN ZOLTÁN

MI7FGD

Miskolc, 2024



SZAKDOLGOZAT FELADAT

Berta István Zoltán

BSc Villamosmérnök jelölt részére

A tervezés tárgyköre: ipari automatizálás

A feladat címe: *Ipari robot tervezése pick and place feladatra labor környezetben*

A feladat részletezése:

1. Mutassa be a technológiát és foglalja össze a fejlesztéssel szemben támasztott követelményeket!
2. Válassza ki és mutassa be a fejlesztés eszközeit!
3. Tervezze meg az irányítórendszert!
4. Valósítsa meg a pick and place feladatra alkalmas robotvezérlést!
5. Foglalja össze a tesztelés során szerzett tapasztalatokat!

Tervezésvezető(k): **Forgács Zsófia**

Konzulens(ek): **Hajdu Botond**

A szakdolgozat kiadásának időpontja: 2024.02.18.

A szakdolgozat beadásának határideje: 2024.05.03.

Miskolc, 2024.05.08.

Dr. Trohák Attila
egyetemi docens
Intézet igazgató

EREDETISÉGI NYILATKOZAT

Alulírott Berta István Zoltán, Neptun-kód: MI7FGD a Miskolci Egyetem Gépészmérnöki és Informatikai Karának végzős villamosmérnök alapszakos hallgatója ezennel büntetőjogi és fegyelmi felelősségem tudatában nyilatkozom és aláírással igazolom, hogy **„Ipari robot tervezése pick and place feladatra labor környezetben”** című szakdolgozatom saját, önálló munkám; az abban hivatkozott szakirodalom felhasználása a forráskezelés szabályai szerint történt.

Tudomásul veszem, hogy szakdolgozat esetén plágiumnak számít:

- szó szerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;
- tartalmi idézet hivatkozás megjelölése nélkül;
- más publikált gondolatainak saját gondolatként való feltüntetése.

Alulírott kijelentem, hogy a plágium fogalmát megismertem, és tudomásul veszem, hogy plágium esetén szakdolgozatom visszautasításra kerül.

Miskolc, 2024. május 3.

.....

Hallgató

Tartalomjegyzék

1.	Bevezetés	3
2.	Az ipari robotok fejlesztési irányai és kihívásai	4
2.1.	Trendek a robotikában	6
2.1.1.	A 2021-es trendek	6
2.1.2.	2022 trendjei	7
2.1.3.	2023 trendjei	7
2.2.	A robotok főbb típusai	8
2.2.1.	A robotok kiválasztása	10
3.	A laborban használt eszközök bemutatása	13
3.1.	A fejlesztéshez használt robotok	13
3.2.	Universal Robot	13
3.2.1.	Universal Robot programozása	14
3.2.2.	Universal Robot kinematikája	16
3.2.3.	Direkt kinematika	17
3.2.4.	Inverz kinematika	18
3.3.	Mitsubishi Robot	18
3.3.1.	Mitsubishi robot kinematikája	20
3.3.2.	Mitsubishi robot programozása	21
4.	Pólusház ellenőrző állomás	23
4.1.	Robot cella tervezése	24
5.	A robot programozása	26
5.1.	Szimuláció létrehozása	26
5.2.	A 3D Monitorban felállított virtuális környezet létrehozása	27
5.3.	Az útvonalpontok	29
5.3.1.	Lineáris és jointos interpoláció a MOV és MVS Utasításokkal	30
5.4.	Robot paraméterek	32

5.4.1.	Mozgási paraméterek.....	32
5.4.2.	Kéz paraméterek (Hand Parameters)	33
5.4.3.	Súly és méret paraméterek (Weight and Size Parameters)	33
5.4.4.	Eszköz paraméterek (Tool Parameters)	34
5.4.5.	Ütközés-észlelési paraméterek (Collision Detection Parameters)	34
5.5.	A robot program.....	35
5.5.1.	Robot vezérlő program	36
5.5.2.	Mágnesező állomás program	42
5.5.3.	Kamera állomás program.....	43
5.5.4.	Bejövő anyagáramlás szimulációja.....	45
6.	Összegzés.....	46
7.	Summary.....	47
8.	Irodalomjegyzék	48
9.	Ábrajegyzék.....	49
10.	Táblázatjegyzék	50

1. Bevezetés

A robottechnológia forradalmasítja az ipari ágazatot, mivel a robotok egyre gyakrabban helyettesítik az emberi munkaerőt. Képesek komplex feladatokat precízen és gyorsan elvégezni, és képesek éjjel-nappal dolgozni, szünet vagy pihenés nélkül. Ez a technológia lehetővé teszi a vállalatok számára a költségek csökkentését, a termelékenység növelését és a munkahelyi biztonság javítását. Ennek eredményeképpen a robotok gyorsan az ipari táj szerves részévé válnak. A robottechnológia az elmúlt években jelentősen fejlődött, lehetővé téve olyan együttműködő robotok kifejlesztését, amelyek hihetetlen koordinációs és precíziós teljesítményre képesek. További fejlesztésekkel a kooperatív robotok felbecsülhetetlen értékű eszközzé lehetnek a vállalkozások és iparágak számára világszerte. A dolgozatom célja, hogy részletesen bemutassam az ipari robotok tervezésének és alkalmazásának folyamatát egy laboratóriumi környezetben, különös tekintettel a pick and place feladatokra.

A dolgozatban bemutatom a technológia fejlődését, a fejlesztési eszközök kiválasztásának folyamatát, valamint a robotvezérlés megvalósításának lépéseit és annak tesztelését. Részletesen elemzem a fejlesztési folyamat során felmerült kihívásokat és az ezekre adott megoldásokat.

A dolgozat második fejezetében áttekintem a robotika alapjait és az ipari automatizálás jelenlegi helyzetét, illetve részletezem az ipari robotok fejlődési trendjeit és a fejlesztés során felmerülő kihívásokat. Tárgyalom a különböző robotok típusait és alkalmazási területeit, valamint az eszközök kiválasztásának folyamatát. A harmadik fejezet az irányítórendszer tervezésére és megvalósítására koncentrálna. Bemutatom a laboratóriumban használt robotokat és perifériákat, valamint részletesen ismertetem a vezérlőrendszer felépítését és működését. A negyedik fejezetben összefoglalom a robotvezérlés tesztelésének eredményeit és a gyakorlati tapasztalatokat. Bemutatom a robotcella tervezésének és megvalósításának lépéseit, valamint a tesztelés során szerzett tapasztalatokat és az ezekre adott megoldásokat. Az ötödik fejezetben a fejlesztés során felmerült kihívásokat és azok megoldásait elemzem.

2. Az ipari robotok fejlesztési irányai és kihívásai

A robottechnológia az elmúlt években jelentősen fejlődött, lehetővé téve olyan együttműködő robotok kifejlesztését, amelyek hihetetlen koordinációs és precíziós teljesítményre képesek. A robottechnológia olyan területeken is alkalmazható, mint például a veszélyes környezetek, ahol az emberek számára túl veszélyes a munkavégzés. A technológia további fejlődésével a robotok ipari alkalmazása valószínűleg még szélesebb körben elterjed majd. Ezáltal értékes emberi erőforrások szabadulhatnak fel más feladatokra. Emellett új foglalkoztatási lehetőségeket is megnyithat az emberek számára, lehetővé téve számukra, hogy magasabb értékű munkákra összpontosítsanak.

Az utóbbi évek számos nehézséget hozott a globalizált ipari termelés terén. A COVID járvány, a szállítási láncok bizonytalanabbá és az energia árak növekedése révén, egyre drágábbá válása a világ fejlett országait az eddigi gyakorlat újragondolására inspirálta. A világ ipari vezető országai mind igyekeznek állami programokkal is segíteni a robot technológia fejlődését. A Nemzetközi Robotikai Szövetség (IFR) 2023 január 12.-én tette közzé éves jelentését „World Robotics R&D Programs” néven.

Kínában a „14. ötévesterv” a robotipar fejlesztésére, az innováció előmozdítására összpontosít. A cél az, hogy Kína globális vezető szerepet töltsön be a robottechnológia és az ipari fejlődés terén. A robotika 8 kulcsfontosságú iparágban szerepel a következő 5 évben, ezen belül Nemzeti Kulcsfontosságú K+F Terv keretében elindult az „Intelligens Robotok” kiemelt program 2022 áprilisában 43,5 millió USD támogatással. Az IFR nemrégiben megjelent „World Robotics” statisztikai évkönyve azt mutatja, hogy Kína elérte a 322 egység robotsűrűséget 10 000 dolgozóra a feldolgozóiparban: az ország 2021-ben az 5. helyen áll a világon, szemben a 2018-as 20. 140 egység értékkel, ezzel a legdinamikusabb fejlődést produkálta

Japánban az „Új robotstratégia” célja, hogy az országot a világ első számú robotinnovációs központjává tegye. A japán kormány 2022-ben több mint 930,5 millió USD támogatást nyújtott. olyan integrált technológiák fejlesztése, amelyek a következő generációs mesterséges intelligencia és robotok magját képezik.

A dél-koreai intelligens robotokra vonatkozó 3. alapterv arra törekszik, hogy a negyedik ipari forradalomban a robotikát a negyedik ipari forradalom központi iparágává alakítsák. A koreai kormány 172,2 millió USD-t különített el az „intelligens robot 2022-es végrehajtási tervéhez”. A "World Robotics" statisztikai évkönyv 2021-ben minden idők

csúcását, 10 000 alkalmazottra jutó 1000 ipari robotot mutatott ki. Ezzel Koreát a legmagasabb robotsűrűséggel rendelkező ország a világon.

A „Horizon Europe” az Európai Unió kulcsfontosságú kutatási és innovációs keretprogramja, amelynek költségvetése 94,30 milliárd USD hét évre (2021-2027). A legfontosabb célok a következők: az EU tudományos és technológiai alapjainak megerősítése, Európa innovációs kapacitásának, versenyképességének és foglalkoztatásának növelése, valamint a polgárok prioritásainak megvalósítása, valamint a társadalmi-gazdasági modellek és értékek fenntartása. Az Európai Bizottság összesen 198,5 millió USD támogatást biztosít a robotikával kapcsolatos 2021–2022-es munkaprogramhoz.

A németországi High-Tech Strategy 2025 (HTS) a német K+F és innovációs program negyedik kiadása. A német kormány 2026-ig évente mintegy 69 millió USD-t biztosít – a teljes költségvetés 345 millió USD öt évre. A HTS 2025 misszió részeként elindult a „Shaping Technology for the People” program. A program célja, hogy a technológiai változásokat a társadalom egészében és a munka világában az emberek javára fordítsa.

Az Egyesült Államokban a National Robotics Initiative (NRI) az Egyesült Államok kormánya által támogatott alapvető robotikai kutatás-fejlesztés céljából indult. Az NRI-3.0 program. A 2021 februárjában bejelentett, integrált robotrendszerrel kapcsolatos kutatásokat folytat, és a korábbi NRI-programokra épít. Az Egyesült Államok kormánya 2021-ben 14 millió USD összeggel támogatta az NRI-3.0 alapot. Ösztönözik az akadémikusok, az ipar, a kormány, a non-profit és egyéb szervezetek közötti együttműködést. A NASA 35 milliárd USD-ral támogatott projektjei olyan kutatási és technológiai fejlesztéseket céloznak meg, amelyek jelentősen növelik a robotok teljesítményét [1].

A 10 000 munkásra jutó robotok számának emelését több ország is kiemelkedő helyen kezeli a programjában. Ennek a fontos mutatónak az alakulását az IFR 2022 december 05-én tette közzé. A kutatási eredmények alapján belátható, hogy a robottechnika terén Ázsia jelentős előnyben van. A nagy népességű országok – Kína és India esetében azt lehetne gondolni, hogy a rengeteg munkaerő miatt nincs szükség sok robotra mégis ebben látják a jövőt és nagy erőfeszítéseket tesznek a fejlődésre [2].

2023. május 3-i adat szerint Az ipari robotok eladása Indiában új rekordot ért el, 54 százalékos növekedés az előző évhez képest. Az éves telepítések számát tekintve India

jelenleg a tizedik helyen áll világszerte. Ezek a World Robotics jelentés megállapításai, amelyet a Nemzetközi Robotika Szövetség (IFR) mutatott be [3].

2.1. Trendek a robotikában

A fejlődés világszerte nagy és gyorsul az IFR Nemzetközi Robotikai Szövetség évente összeállítja az öt legfontosabb rendet, ami meghatározza a robotipart.

2.1.1. A 2021-es trendek

1. Intelligens robotok

Mesterséges intelligencia és érzékelőrendszerek kombinációjával a robotok képesek nehezebb feladatokra, mint például szemétválogatás, és könnyebben telepíthetők, programozhatók, valamint integrálhatók az automatizálásba és az Ipar 4.0 stratégiákba.

2. A robotok intelligens gyárakban dolgoznak

Az autóipar az intelligens gyártásban úttörő szerepet játszik, ahol robotok és autonóm mobil robotok hálózatos interakciója teremti meg a rugalmasságot és a gyors adaptációt az összeszerelősorokon.

3. A robotok új piacokat hódítanak meg

Az automatizálás egyre inkább teret nyer az olyan iparágakban is, mint az élelmiszer- és italgyártás, textilipar, fatermékek és műanyagok gyártása, új üzleti modellek létrehozásával és a hagyományos gyártósorok elavulásával.

4. A robotok csökkentik a szénlábnyomot

Az energiahatékony robotok és a nagyobb precizitású gyártás kevesebb energiafelhasználást és kevesebb nem megfelelő termék előállítását eredményezik, pozitív hatással az erőforrás-inputra és a kibocsátásra.

5. A robotok segítenek az ellátási láncok biztonságossá tételében

Az automatizálás növeli a gyártók rugalmasságát és biztonságát, lehetővé téve számukra, hogy kevésbé fűgjenek a globalizált ellátási láncoktól, különösen a magas költségekkel járó régiókban [4].

2.1.2. 2022 trendjei

1. Új iparágak által elfogadott robotok

Az automatizálás viszonylag új szegmensei gyorsan alkalmazzák a robotokat. Az e-kereskedelem tovább gyorsul, és új robotok telepítése történik olyan területeken, ahol korábban nem léteztek. A munkaerőhiány miatt egyre több vállalat fordul az automatizálás felé, ideértve a kiskereskedelmet és az éttermeket is. Új iparágak, mint a szállítás és logisztika, profitálnak a technológiai fejlődésből.

2. A robotok könnyebben használhatók

Az új generációs robotok egyszerűbben programozhatók és irányíthatók. Az egyszerű ikonvezérelt programozás és a felhasználóbarát felületek irányába mutató tendencia jelentős könnyebbséget jelent a robotok bevezetésében.

3. Robotok és emberek készségfejlesztése

Egyre több kormány és vállalat felismeri a robot- és automatizálási oktatás fontosságát. Az adatvezérelt gyártás területén az oktatás és képzés kap nagyobb hangsúlyt, beleértve a külső oktatási útvonalakat is.

4. A robotok biztosítják a termelést

A kereskedelmi feszültségek és a COVID-19 közelebb viszik a gyártást az ügyfelekhez. Az automatizálás segíti a vállalatokat az üzleti életbe való visszatérésben, az Egyesült Államokban például a robotrendelések jelentős növekedése tapasztalható.

5. A robotok támogatják a digitális automatizálást

Az adatok fontossága kiemelkedő a jövő gyártásában. Az intelligens automatizálás és a mesterséges intelligencia összekapcsolása lehetővé teszi a vállalatok számára az adatok elemzését és az okos döntéshozatalt. A robotika és mesterséges intelligencia egyre szorosabban kapcsolódik össze, ami a technológiai fejlődés újabb mérföldkövéhez vezet.
[5]

2.1.3. 2023 trendjei

A 2023-ban közzétett trendekben van a korábbi években már leírt trend is, de ezek egymás épülnek, és fontosságuk még hangsúlyosabb. Az 5 legjobb robottrend 2023-ban
© International Federation of Robotics.

1. Energiahatékonyság

A robotika elterjedése jelentősen hozzájárul az energiahatékonysághoz a gyártásban. A robotok alacsonyabb energiával működnek és növelik a termelés hatékonyságát, miközben csökkentik a fűtési költségeket. Az energiatakarékos

technológiák segítségével a robotok képesek visszatáplálni az elektromos hálózatba, jelentős energiamegtakarítást eredményezve.

2. Reshoring

Az ipari vállalatok egyre inkább visszatelepítik gyáraikat az ügyfelek közelébe a rugalmasság és az ellátási lánc biztonságossága érdekében. Az akkumulátorok és mikrochipgyártás robotizálása fontos szerepet játszik ebben a trendben, támogatva az ipari folyamatok hatékonyabbá tételét.

3. A robotok könnyebben használhatók

A robotprogramozás egyre egyszerűbbé válik, és az új technológiák lehetővé teszik a nem szakértők számára is a robotok kezelését. Az intuitív felhasználói felületek és az alacsony kódú vagy akár kód nélküli programozási lehetőségek újabb dimenziókat nyitnak meg a robotika területén.

4. Mesterséges intelligencia (AI) és digitális automatizálás

Az AI és a digitális technológiák fejlődése átalakítja a gyártást, miközben segíti a vállalatokat a változó környezet kezelésében. Az AI-alapú robotok lehetővé teszik a gyors és költséghatékony megoldásokat a gyártási folyamatok optimalizálásában és a rugalmasság növelésében.

5. Az ipari robotok második élete

Az ipari robotok hosszú élettartama lehetővé teszi számukra a "második életet". A robotgyártók speciális javítóközpontokat működtetnek, hogy felújítsák és frissítsék a használt robotokat, hozzájárulva a körforgásos gazdasághoz és a költségmegtakarításhoz [6].

2.2. A robotok főbb típusai

A robot egy újraprogramozható, többfunkciós manipulátor, amely anyagok, alkatrészek, szerszámok vagy speciális eszközök mozgatására lett tervezve, változó programozott mozgásokon keresztül, különböző feladatok elvégzésére.[14]

Az ipari robotoknak három fő alkotóeleme van:

- 1.egy vagy több kar, amelyek általában egy fix alapra vannak helyezve és több irányba képesek mozogni;
- 2.egy manipulátor, a robot "keze", amely a szerszámot vagy a megmunkálandó alkatrészt tartja;
- 3.egy vezérlő, amely részletes mozgásutasításokat ad.

Az ipari robotkarok vannak jelen legrégebben ezek a gépek a legmegbízhatóbbak a piacon és általában sokkal robusztusabbak, mint az újabb másfajta robotok.

Számos ipari robotot ismerünk manapság, de a legáltalánosabb típusok az alábbiak:

- Csuklós ipari robotok
- Kollaboratív ipari robotok (kobotok)
- SCARA ipari robotok
- Delta ipari robotok
- Ortogonális vagy portál ipari robotok
- Kétkarú ipari robotok

Az ipari robotok csoportosítása a művelet-beviteli módszereik szerint:

- Visszajátszó ipari robotok
- Szekvencia ipari robotok
- Numerikus vezérlésű (NC) ipari robotok

A csuklós robotok olyan mozgást tesznek lehetővé, amelyek hasonlítanak az emberi váll, könyök és csukló mozgásaira. Ezek a robotok 4-, 5-, 6- vagy akár 7-tengellyel rendelkezhetnek. Gyakran használják hegesztéshez és festéshez, amelyek az ipari robotok jelentős részét képezik. Nagyon sokoldalúak és különböző környezetekben alkalmazhatók, beleértve a logisztikai központokat és az alkatrészfeldolgozó üzemeket is.

A csuklós robotok mozgástartománya növekszik a tengelyek számának növekedésével, és képesek szűk helyeken is működni, ha a kar hajlik. Általában fixen telepítették őket, és nincsenek felszerelve környezeti érzékelőkkel, ezért a munkaterületüket el kell választani a környezetüktől például rácsokkal.

Ezek a robotok nagy sebességűek, precízek, újra programozhatók és biztonságosak, de tanításukhoz és telepítésükhöz szakember szükséges. Mivel fixen telepített gépek, nagyobb térre van szükségük. A gyors termelési ciklusokkal dolgozó vállalatok számára a hagyományos robotok lehetnek megfelelőbbek, mivel gyorsabbak. Különösen gyakori az autógyártásban.

A kooperatív robotok (kobot) az utóbbi évek technológiai fejlődésének eredményei. Ezek könnyebben használhatók, integrált munkabiztonságot kínálnak, és együttműködnek az emberekkel. Rugalmasabbak és könnyebben mozgathatóak, és általában nincs szükség külső szakemberekre a programozáshoz [7]. Kisebb helyet foglalnak el, és biztonságosabban működnek emberek közelében, bár lassabbak lehetnek. A kobotok könnyen csatlakoztathatók a gépvezérlőkhöz és más technológiákhoz, és

rugalmasan használhatók különböző feladatokhoz. A választás a feladat igényeitől és a környezeti tényezőktől függ.

2.2.1. A robotok kiválasztása

Egy robot kiválasztása függ a helyszíntől, a termelési céloktól és személyes igényeitől is. Az érzékelési tartomány, a teherbírás, a mozgás típusa, a bővíthetőség és az interakció olyan technológiákkal, mint a mesterséges intelligencia (AI) vagy a képfeldolgozás, mind befolyásolják azt, hogy melyik robot lesz a legmegfelelőbb.

Fontos kérdések egy robot kiválasztásakor:

- Milyen termékek gyártására vagy feldolgozására használná a robotot, és milyen módszerekkel?
- Milyen mozgásokat kell elvégeznie a robotnak a feladatok végrehajtásához?
- Milyen távolságot kell megtennie a robotnak a munkaterületén?
- A robotot házon belül képzett szakemberek fogják irányítani, vagy kívülről kezelik majd?
- Milyen típusú alkatrészgyártásra és kapacitásra van szükség, figyelembe véve a megismételhetőséget és a pontosságot?
- Milyen nagyságú beruházást igényel a robot telepítése és működtetése?
- Várhatóan mennyi idő alatt térülne meg a robotra való beruházás? [8]

Buri Zs. (2022) elemzésének segítségével részletesen összehasonlítottam négy különböző kobotot: Universal Robots UR5e, KUKA LBRIIWA 7, Techman Robot TM5-900 és ABB GoFa CRB 15000. Az összehasonlítás során számos kritériumot vettem figyelembe, amelyeket az Analytical Hierarchy Process (AHP) módszerrel rangsorolt. Ez a módszer lehetővé teszi a komplexrendszerek és multikritériumos döntések alapos vizsgálatát.

Az AHP alapú rangsorolás során számos tényezőt értékelt, mint például az árat, a maximális teherbírást, a vezérelt tengelyek számát, a karkinyúlást, a pozíció ismételhetségét, a súlyt, az IP védelmi szintet, a mozgási sebességet és az energiafogyasztást. Ezek a tényezők mind fontosak lehetnek egy kobot kiválasztásánál, hiszen befolyásolják a robot teljesítményét és alkalmazhatóságát.

Az AHP módszer eredményeinek alapján a Kesselring módszert alkalmazta az alternatívák sorrendjének meghatározására. Ez azért fontos, mert az emberek képessége az információ feldolgozására korlátozott, ezért az értékelést minden kritérium esetében külön kellett végezni. A Kesselring módszer alkalmazásával lehetőségem volt különböző,

de valamilyen arány- vagy intervallumskálán mérhető műszaki paraméterek összehasonlítására, ami lehetővé tette a komplex rendszerek elemzését [9].

Az összehasonlítás eredményeként az ABB GoFa CRB 15000 kapta a legmagasabb pontszámot. Annak ellenére, hogy magasabb az ára, mégis a legjobb pontossággal, gyorsasággal és kinyúlási mérettel rendelkezik a kínált lehetőségek közül. Ennek megfelelően, ha a felhasználó számára a gyorsaság és a pontosság kiemelten fontos, az ABB GoFa lehet a legmegfelelőbb választás.

1. táblázat Robotparaméterek táblázatban [9]

		Universal Robots UR5e		KUKA LBR IIWA 7		Techman Robot TM5-900		ABB GoFaCRB 15000	
#	Kritérium	Érték	Pont	Érték	Pont	Érték	Pont	Érték	Pont
1	Ár	28,750 €	3	62,500€	1	27,000 €	4	31,500€	2
2	Max. teherbírás	5 kg	3	7 kg	4	4 kg	2	5 kg	3
3	Vezérelt tengelyek száma	6	3	7	4	6	3	6	3
4	Max. kinyúlás	850 mm	2	800 mm	1	900 mm	3	950 mm	4
5	Pozíció ismétlés	0.1 mm -0.03 mm	2	+/-0.1 mm	4	+/-0.05 mm	3	+/-0.05 mm	3
6	Súly	20.6 Kg	4	23 Kg	2	22.6 Kg	3	27 Kg	1
7	Lábméret	149 mm	3	136 mm	4	152 mm	2	165 mm	1
8	Biztonság	IP54	4	IP54	4	IP54	4	IP54	4
9	Mozgási sebesség	1000 mm/s	1	1500 mm/s	3	1400 mm/s	2	2200 mm/s	4

A bemutatott példa jól szemlélteti, hogy a döntéshozatal komplexitása és a lehetséges szempontok száma rendkívül nagy kihívást jelent. Azonban az ipari robottechnológia és a hozzá kapcsolódó mesterséges intelligencia által nyújtott megoldások segítségével válaszolhatunk a negyedik ipari forradalom gyorsan változó világának követelményeire.

A hagyományos ipari robotok elsősorban nagy súlyok nagy sebességgel és pontossággal történő manipulálására alkalmasak. Ezzel szemben a kobotok kiválóan használhatók anyagmozgatásra, rakodásra, palettázásra, csomagolásra és tesztelésre, valamint a rugalmasságuk révén az ipar új területein is megállják a helyüket, ahol a gyorsaság mellett a rugalmasság is kiemelten fontos.

3. A laborban használt eszközök bemutatása

3.1. A fejlesztéshez használt robotok

A laborban különböző típusú robotok találhatóak. Ezek lehetnek tréningcélú robotok, gyártásra előkészített robotok. A gyártásra előkészített robotok a sorozatgyártás előtti tesztelési és kalibrálási fázisban vannak, ahol különböző paramétereket és funkciókat ellenőriznek annak érdekében, hogy a gyártási folyamatban zökkenőmentesen működjenek.

3.2. Universal Robot

A laborban általában mindig van egy-két Universal robot különböző okokból. Leggyakrabban egy UR5 típusú, 6 tengelyes, 850 mm kinyúlású robot található itt. Az "5" a névben a robot maximális terhelhetőségét jelöli kilogrammban. Ez nem csak a megfogót, hanem a manipulált munkadarabot is magában foglalja.



1. ábra Universal robot UR5 [10]

Az UR5 az egyik legelterjedtebb kooperatív robot. A robot megfelel az EN ISO 13849:2008 PL d és az EN ISO 10218-1:2011 5.4.3 szabványoknak, köszönhetően biztonságosságának és könnyű kezelhetőségének. Néhány biztonsági sík és lézerszkenner alkalmazásával akár biztonsági ketrec nélkül is biztonságosan üzemeltethető. Ez a típus kiválóan alkalmas kisebb tárgyak precíz mozgására és emberek melletti munkavégzésre. Ugyanakkor nehezebb tárgyak gyors, szünetmentes mozgására, illetve olyan munkakörnyezetben, ahol a robotkarnak nagy erőt kell kifejtenie, nem megfelelő.

2. táblázat UR5 Specifikációk [10]

Terhelhetőség	5 kg / 11 lbs
Kinyúlás	850 mm / 33.5 in
Szabadsági fokok	6 forgó csukló
Programozás	Polyscope grafikus felhasználói felület 12” érintőképernyőn

3. táblázat UR5 Mozgási paraméterek [10]

Tengelymozgás robotkar	Munkatartomány	Maximális sebesség
Alap	± 360°	± 180°/mp
Váll	± 360°	± 180°/mp
Könyök	± 360°	± 180°/mp
Csukló 1	± 360°	± 180°/mp
Csukló 2	± 360°	± 180°/mp
Csukló 3	± 360°	± 180°/mp
Tipikus eszköz	1 m/másodperc	39.4 in/másodperc

Az Universal Robot UR5 beállítása rendkívül egyszerű és gyors, ami jelentősen megkönnyíti a használatát. A robot könnyen megtanulható, így karbantartás során is könnyedén ellenőrizhetők a különböző pozíciók. Az UR5 robot képes Freedrive üzemmódban működni, amely lehetővé teszi a manuális mozgatót. Ebben az üzemmódban a felhasználó egyszerűen megragadja a robotot és megnyomja a Freedrive gombot, így kézzel irányíthatja annak mozgását. Ez a funkció különösen hasznos a pontos beállítások elvégzéséhez és a robotprogramok finomhangolásához.

3.2.1. Universal Robot programozása

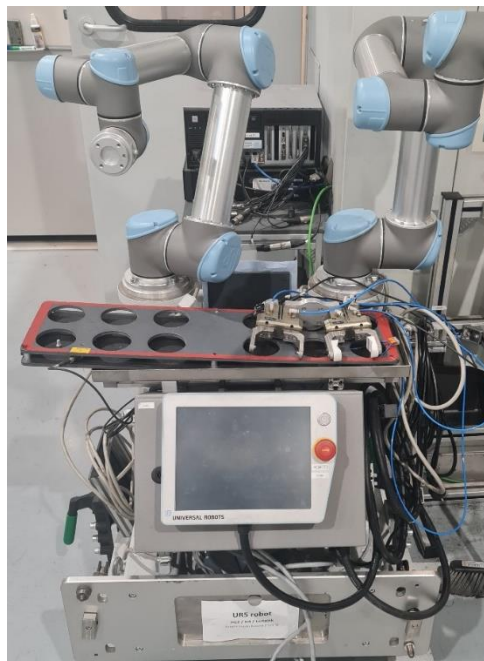
Az Universal robotok mindegyikéhez tartozik egy vezérlődoboz, amely a robot minden funkcióját kezeli, ideértve az I/O, TCP/IP 100 Mbit: IEEE 802.3u, 100BASE-TX Ethernet csatlakozót és a Modbus TCP-t. Az UR5 robot vezérlőszekrényén 16 digitális bemenet/kimenet (DI/DO) és 2 analóg bemenet/kimenet (AI/AO) található. A vezérlőszekrény részét képezi egy vezérlőpanel is, amely egy 12”s érintőkijelzőből áll.

4. táblázat UR5 I/O Portok [10]

I/O portok	
Digitális bemenet	16
Digitális kimenet	16
Analóg bemenet	2
Analóg kimenet	2
I/O tápellátás	24V 2A
Kommunikáció	TCP/IP 100Mbit, Modbus TCP, Profinet, EthernetIP

A robotot általában az érintőkijelzőn keresztül lehet vezérelni és programozni. Az UR robotok egy ingyenes szimulációs környezetet is biztosítanak, amely hasonlóan működik a robot szoftveréhez, de nem bizonyult túl hatékonynak, mivel hiányoztak belőle kulcsfontosságú elemek, amelyek a szimulációt és a programfejlesztést segítették volna.

A programozás Polyscope rendszerben történik, amely lehetővé teszi a felhasználók számára, hogy "drag and drop" módszerrel állítsanak össze programokat. Ez alapvetően csökkenti a robotprogramozási ismeretekre vonatkozó előfeltételeket. A Polyscope a robotkarokon futó beépített szoftver, amely a programok létrehozását, tesztelését és végrehajtását teszi lehetővé.



2. ábra Általán átalakított robot tesztelő/fejlesztő állomás

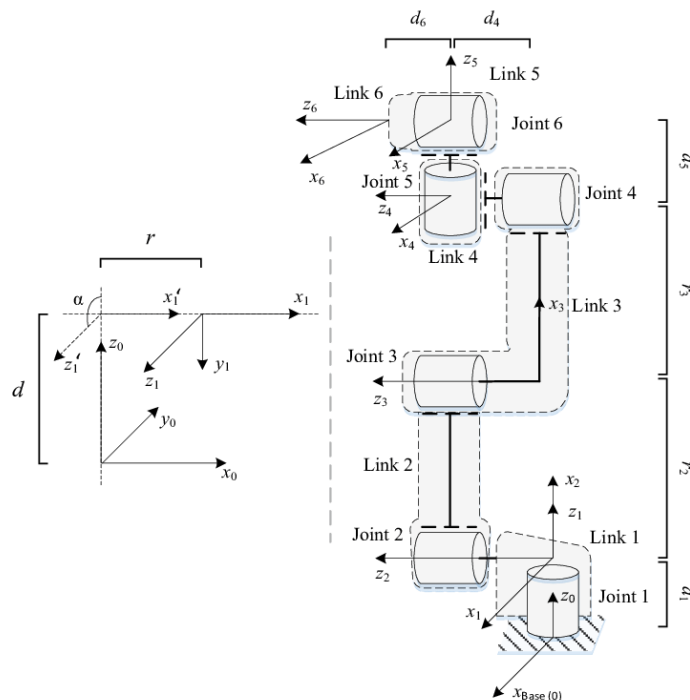
Bár a Polyscope nagyon felhasználóbarát a legtöbb alapvető alkalmazás esetén, a UR robotok programozása magában foglal egy erősebb, szkript alapú programozási lehetőséget is, amelyet URScript-nek hívnak. A URScript egy Python-szerű nyelv, amely lehetővé teszi a felhasználók számára, hogy fejlettebb logikát és funkciókat építsenek be a robotműveleteikbe. Ezzel a nyelvvvel a felhasználók képesek változókat kezelni, ciklusokat és feltételes utasításokat használni, valamint integrálni a robotkarokat külső eszközökkel és rendszerekkel.

URCaps egy másik innovatív megoldás az Universal Robots részéről, amely egy bővítő platformot kínál a UR robotkarok számára. Ez a rendszer lehetővé teszi a felhasználók számára, hogy személyre szabott bővítményeket, úgynevezett "URCaps" segítségével bővítsék a robot képességeit. Ezen bővítmények Java nyelven íródhatnak, és közvetlenül integrálhatók a robot vezérlőrendszerébe.

URCaps különösen hasznos olyan speciális alkalmazások esetében, ahol a standard funkciók és szkriptek nem elegendők. Például, egy URCap lehet egy szoftverbővítmény, amely egy speciális kamera vagy más látórendszer integrálását teszi lehetővé a robotkarba, így lehetővé téve a robot számára, hogy bonyolultabb és érzékenyebb feladatokat is elvégezzen, mint például minőségellenőrzés vagy alkatrészek pontos azonosítása.

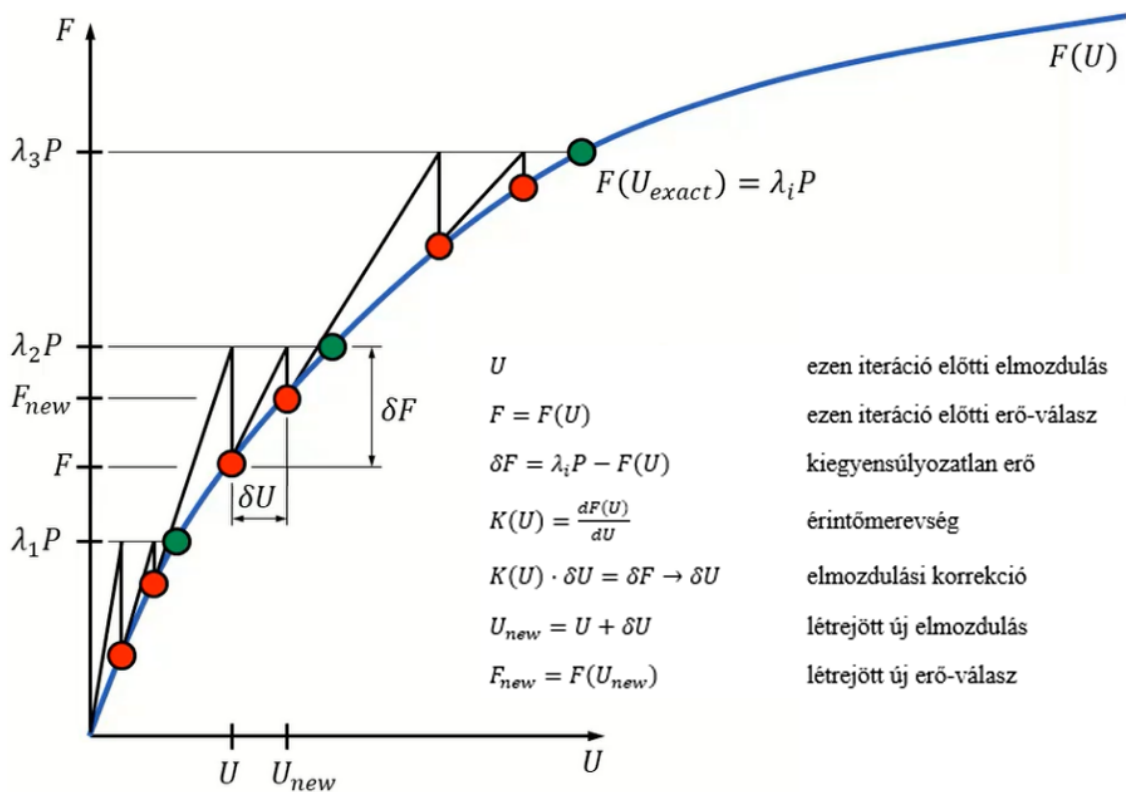
3.2.2. Universal Robot kinematikája

A UR5 egy hat szabadságfokkal rendelkező (DOF), artikulált robotkar, ami azt jelenti, hogy hat forgócsuklója van, mindegyik egy-egy független mozgási szabadságot biztosít. Ez a konfiguráció lehetővé teszi a kar számára, hogy háromdimenziós térben szinte bármilyen pontba és orientációba elérjen. A robotkar kinematikája a Denavit-Hartenberg (D-H) paraméterezési módszerrel modellezhető, amely egy szabványos megközelítés a csuklók és a karok geometriai kapcsolatának leírására.



3. ábra UR5 robot paraméterek a Denavit-Hartenberg módszer szerint [11]

A robot csuklóit a bázistól (a felszerelési ponttól) számítsuk. A csuklók sorba vannak kötve minden csukló az előzőtől kapja a kommunikációt és attól számítsa, aki a pozícióját egészen a Tool Mountig Bracket-ig. A robot első 4 csuklója egy síkba esik egymással ami egyszerűsíti a felépítést de figyelni a kell a szingularitás pontokra. A szingularitás pontok azok a helyzetek amikor a robot csuklói egy síkba esnek egymással ilyenkor ahhoz, hogy a robot Tool Center Point-ja egy kis pozíció változtatást tegyen ahhoz a többi csuklónak egye nagyobb mozgást kell megtennie például a 3. csuklónak végtelenül gyorsan kéne mozognia.



4. ábra Newton-Raphson iterációs eljárás [12]

3.2.3. Direkt kinematika

A direkt kinematika feladata, hogy adott csuklózögek esetén meghatározza a robotkar végpontjának (end-effector) helyzetét és orientációját a térben. Ezt a számítást a D-H paraméterek felhasználásával végzett transzformációs mátrixok segítségével végezzük, lásd 3. ábra. A transzformációs mátrixok sorozata, amely minden egyes csuklóra vonatkozik, összegezhető, hogy megkapjuk a robotkar teljes helyzetét.

3.2.4. Inverz kinematika

Az inverz kinematika talán a kinematikai számítások legkritikusabb része, mivel itt a cél a robot végpontjának kívánt helyzetéből és orientációjából kiindulva meghatározni a szükséges csuklósögeket. Ez a folyamat általában bonyolultabb, mert több lehetséges megoldás is létezhet, vagy akár nincs is megoldás bizonyos pozíciók esetén. A UR5 esetében speciális algoritmusokat és numerikus módszereket, mint például a Newton-Raphson módszert használják az inverz kinematikai egyenletek megoldására, lásd 4. ábra.

3.3. Mitsubishi Robot

A munkahelyemen, ahol több különböző gyártási és összeszerelési folyamat zajlik, számos helyen alkalmazunk Mitsubishi robotokat. Ezek a robotok kritikus szerepet töltenek be az automatizált gyártósorokon, ahol a precíziós és ismételhetőség elengedhetetlen. A Mitsubishi robotokat főként azért választottuk, mert képesek megfelelni a gyors tempójú, változatos munkakörnyezet kihívásainak, melyek közé tartozik a komplex összeszerelési feladatok elvégzése.

A Mitsubishi RV-2FR-R robotja a vállalat legújabb fejlesztései közé tartozik, amely az ipari automatizálás területén kíván új megoldásokat nyújtani. Ez a robot különösen alkalmas összetett feladatok ellátására, mint például precíziós szerelési munkálatok, minőségellenőrzés és kis alkatrészek kezelése. A robot technológiai újításainak és moduláris felépítésének köszönhetően jelentősen növelhető a gyártási folyamatok hatékonysága és rugalmassága.



5. ábra Mitsubishi Labor Tréning Robot

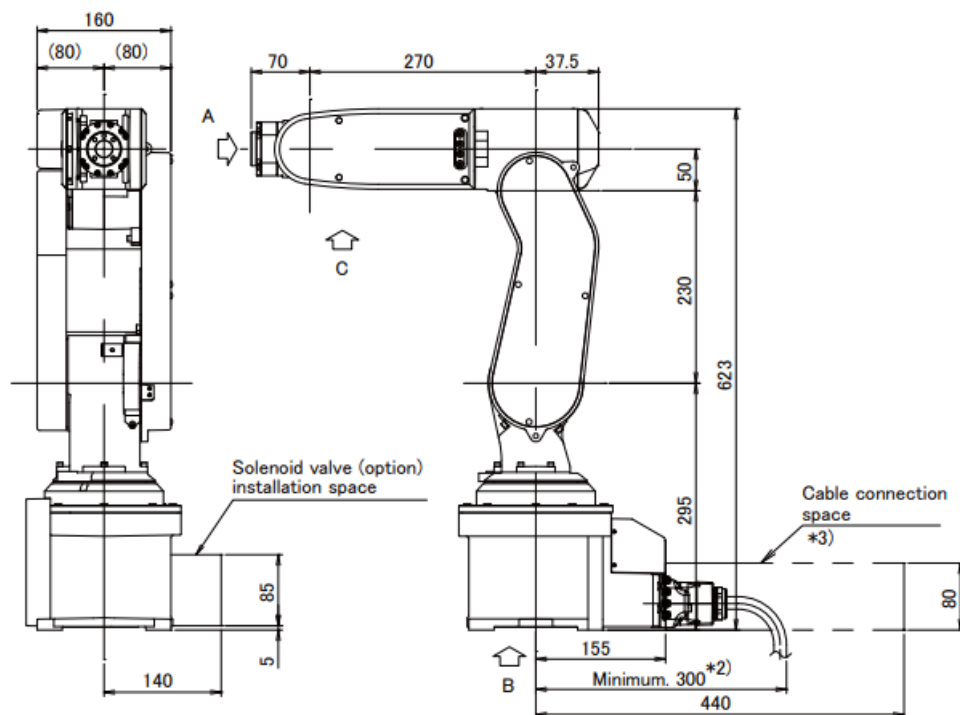
Az RV-2FR-R robot a Mitsubishi Electric által fejlesztett hat szabadságfokkal rendelkező vertikális robot, amely különösen alkalmas különböző ipari alkalmazásokra. A robot kinematikai felépítése lehetővé teszi a nagyfokú rugalmasságot és pontosságot az automatizált műveletek során.

5. táblázat Mitsubishi RV-2FR Specifikáció [13]

Terhelhetőség	2 kg
Tengelyek száma	6
Hatósugár	504 mm
Maximális sebesség	4955 mm/s
Ismétlési pontosság	$\pm 0,02$ mm
Tömeg	19 kg
Ciklus idő	0,6s
Vezérlő típus	CR800-D

Az RV-2FR-R robot a Mitsubishi Electric által fejlesztett speciális szervomotorokkal és meghajtó rendszerekkel van felszerelve, amelyek magas forgatónyomatékot és gyors forgási sebességet (4955 mm/s) biztosítanak. Az RV-2FR-R robot további műszaki jellemzői közé tartozik a karok terhelhetősége, amely 2 kg

maximális súlyt képes kezelni, valamint a karok maximális kinyúlása, amely 504 mm. A robot ismétlési pontossága ± 0.02 mm, ami a robotika területén kiemelkedőnek számít, és lehetővé teszi a rendkívül precíz munkavégzést. Ezek a motorok és meghajtók kiválóan alkalmasak folyamatos működésre, ami lehetővé teszi a robot számára, hogy rövid ciklusidők mellett is magas produktivitást érjen el. A robotkarok nagy merevségűek, ami növeli a gép teljesítményét és pontosságát kritikus alkalmazások esetén, mint például precíziós összeszerelés vagy minőségellenőrzést.



6. ábra RV-2FR befoglaló méretei [13]

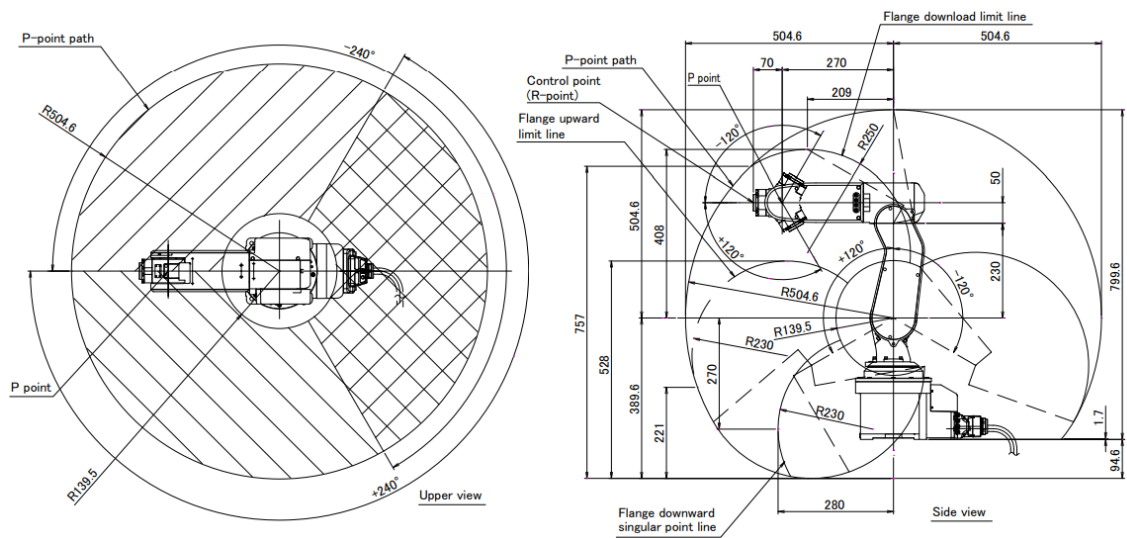
A robot különböző munkakörnyezetekbe és gyártósorokba való beilleszkedését a széles körű interfészek és csatlakozási lehetőségek támogatják. A robot képes többek között pneumatikus és elektromos rendszerekkel közvetlenül kapcsolódni, ami megkönnyíti a különböző műveleti egységek, mint például 3D látórendszerek vagy automatizált anyagmozgató rendszerek, integrálását.

3.3.1. Mitsubishi robot kinematikája

Az RV-2FR-R hat tengelye (J1-J6) különböző mozgásokat tesz lehetővé, amelyek kombinálásával a robot képes bonyolult feladatokat is elvégezni:

- J1 (Bázistengely 480°): Ez a tengely teszi lehetővé a robot alapjának teljes körű forgását. Az alap forgása alapvető a munkaterület kiterjesztéséhez és a robot által elérhető pozíciók számának maximalizálásához.

- J2 és J3 (Kartengelyek J2:240° J3:160°): Ezek a tengelyek a robot karjának fő mozgását biztosítják, lehetővé téve a kar emelését és süllyesztését. Ezek a mozgások nélkülözhetetlenek a munkadarabok eléréséhez különböző magasságokban.
- J4 és J5 (Csuklótengelyek J4:400° J5:240°): A robot csuklóinak mozgása döntő a munkaeszközök (például fogók) orientációjának finomhangolásában. Ezek a tengelyek biztosítják, hogy a robot képes legyen az eszközök pontos beállítására a különböző műveletek során.
- J6 (Forgótengely J6:720°): Az eszközök, mint például fogók vagy hegesztőpisztolyok, végleges orientációjának beállítása. Ez a tengely kritikus a műveletek végrehajtásának pontosításában, különösen összetett összeszerelési feladatok esetén.



7. ábra Mitsubishi RV-2FR-R Mozgása [13]

3.3.2. Mitsubishi robot programozása

A robotok programozása általában az MELFA Basic IV, V vagy VI, illetve a MELFA Works programozási nyelveken keresztül történik. Ezek a nyelvek a BASIC programozási nyelv szintaxisát követik, amely elősegíti a könnyű elsajátítást és alkalmazást a robotikai programozás területén.

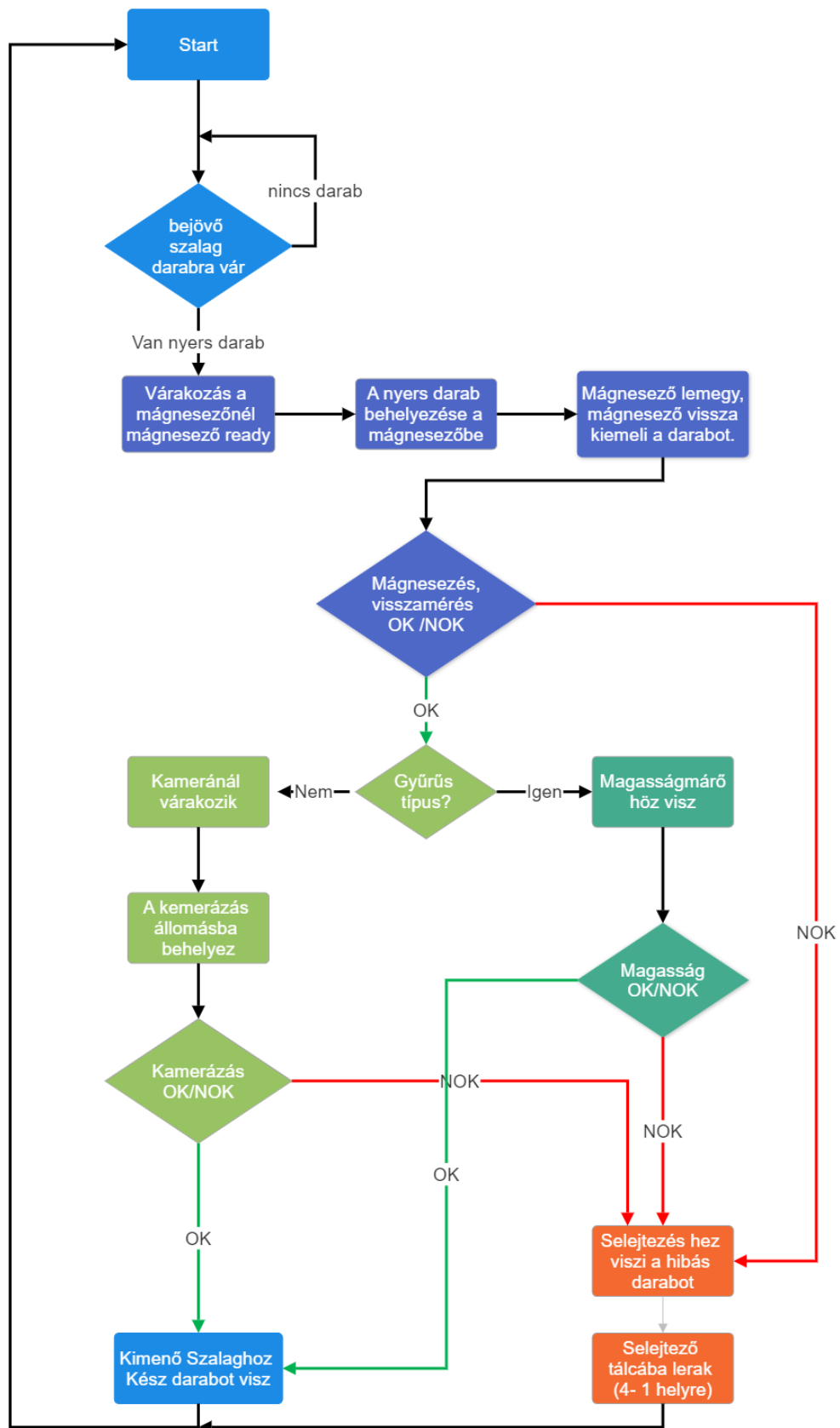
A Mitsubishi által biztosított RT ToolBox2 és RT ToolBox3 szoftverek átfogó támogatást nyújtanak a robotprogramok tervezésében, szimulációjában és monitorozásában. Ezek a szoftverek lehetővé teszik a felhasználók számára, hogy

tervezzék a robotpályákat, szimulálják a programokat, és kommunikáljanak a robot vezérlőjével.

A programozási folyamat során kritikus jelentőségű a biztonsági előírások betartása. Ez magában foglalja a vészleállító rendszerek beállítását, a mozgási korlátok definiálását és a rendszeres karbantartási protokollok követését a működési biztonság és a balesetek megelőzése érdekében.

Összefoglalva, a Mitsubishi robotok programozása magas szintű technikai és analitikai készségeket igényel az automatizációs technológiák területén. A folyamatos képzés és gyakorlati tapasztalat elengedhetetlen a hatékony és biztonságos robotprogramok kifejlesztéséhez.

4. Pólusház ellenőrző állomás



8. ábra Az ellenőrzés folyamata

A robotcella feladata a pólusházak kezelése és minőségi ellenőrzése. A munkám célja fontos gyártási lépés a pólusház ellenőrzéséhez a kiválasztott robot segítségével az általam írt programmal egy gyorsabb, költséghatékonyabb munkaállomás megtervezése. Kulcsfontosságú szerepet tölt be az elektromos motorok gyártási folyamatában. Ennek a célja, hogy biztosítsa a pólusházak megfelelnék az előírt minőségi követelményeknek, ami elengedhetetlen a motorok megbízható működéséhez.

Az 8. ábrán látható részletes folyamatleírás bemutatja a robotcella működésének lépéseit, kezdve a nyers pólusházak futószalagról történő felvételével, a mágneses tulajdonságokkal való ellátáson át, egészen az ellenőrzött darabok kimeneti futószalagra helyezéséig.

Az állomás első lépéseként egy robotkar emeli le a pólusházakat a futószalagról. Ezt követően a darabokat egy mágnesező egységbe helyezi, ahol a korábban nem mágneses ferrit mágneseket az állórészben fel mágnesezik. A mágnesezés folyamatát és a mágnesek jellemzőit — mint az erősség és a polaritás — egy magnetométerrel ellenőrzik.

A mágnesek ellenőrzése után a robot átmozgatja a munkadarabokat a mágnesezőből, vagy a kamera vizsgálati állomásába, vagy a magasságellenőrző egységbe. Amennyiben a munkadarab gyűrűs típusú, úgy a kamerás vizsgálat nem lehetséges, mert a gyűrű eltakarja a mágneseket. Ebben az esetben egy magasságmérőt alkalmazunk, amely megméri a gyűrű helyzetét a pólusházban, és amennyiben ez vízszintes, úgy megállapítható, hogy a mágnesek helyes pozícióban vannak. Nem gyűrűs típusok esetében a kamera két oldalról telecentrikus lencsével fényképezi le a mágneseket, és ez alapján hoz döntést a mágnesek pozíciójáról.

Az ellenőrzési folyamat után az elfogadott munkadarabokat a robot a kimeneti futószalagra helyezi, míg azokat, amelyek nem felelnek meg a szigorú minőségi követelményeknek, selejtdobozba helyezi, így biztosítva a gyártási folyamat integritását.

4.1. Robot cella tervezése

A robot kiválasztásánál a több lehetőségem is volt és miután összehasonlítottam akt robot előnyeit és hátrányait a Mitsubishi robot mellett döntöttem. Az alábbi adatok bemutatják, miért választottam a Mitsubishi modellt:

- **Pontosság:** A Mitsubishi RV-2FR ismert kiemelkedő pontosságáról, amely +/- 0.02 mm-es ismétlési pontosságot biztosít, szemben az UR5 modell +/- 0.1 mm-es pontosságával.
- **Szimulációs képességek:** A Mitsubishi RT Toolbox 3 szoftver komplex szimulációs lehetőségeket kínál, amelyek lehetővé teszik a robotcella teljes körű tesztelését még a valós beüzemelés előtt. Ez a fajta szimuláció fontos a hibák előzetes azonosításában és a folyamat optimalizálásában. Az UR5 szoftvere is kínál szimulációs lehetőségeket, azonban ezek kevésbé részletesek, ami korlátozhatja a komplexebb alkalmazások fejlesztését.
- **Sebesség és terhelhetőség:** A Mitsubishi RV-2FR gyorsabb ciklusidőket kínál (terhelhetőség akár 2 kg-ig), ami növeli a gyártási kapacitást és hatékonyságot. Ezzel szemben az UR5 modellel, amely maximum 5 kg terhelhetőséget kínál, de alacsonyabb sebességgel működik, ami lelassíthatja a termelési folyamatokat.

Az UR5 modell ugyan széles körben elismert a rugalmasságáért és könnyű programozhatóságáért, de a projekt specifikus igényei – különösen a nagy pontosságú és megbízható műveletek – jobban teljesültek a Mitsubishi robot alkalmazásával. A választás így nemcsak technikai előnyöket hozott, hanem biztosította, hogy a gyártási folyamat során a lehető legkisebb legyen a selejtszázalék, ami kulcsfontosságú volt a termelési költségek optimalizálása szempontjából.

A robot precíziós működését tovább növeltem egy speciálisan tervezett pneumatikus SCHUNK megfogó integrálásával, amely lehetővé tette a pólusházak pontos és stabil kezelését a gyártási folyamat során.

A fejlesztés során a Mitsubishi RT Toolbox 3 programot alkalmaztam, amely kiválóan támogatta a különböző munkaállomások és a robot közötti interakciók szimulálását.

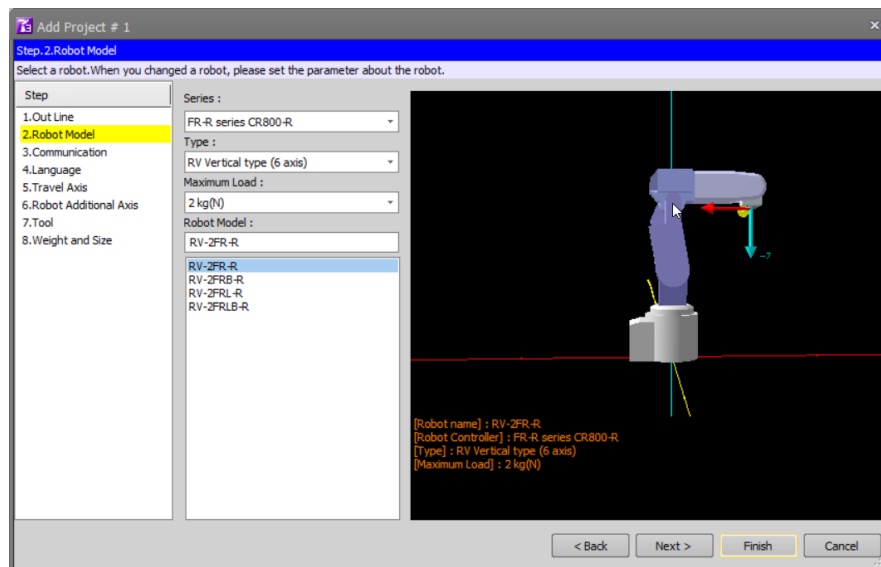
5. A robot programozása

A robot programozása során két fő lehetőség áll rendelkezésünkre: használhatjuk a vezérlőhöz csatlakoztatott kézi vezérlőpanelt, amely saját kijelzővel és billentyűzettel rendelkezik, vagy alkalmazhatjuk a számítógépet a programozási folyamat során. Az én esetemben az RT Toolbox 3 programot választottam a robot programozásához, mivel ez a szoftver rendkívül fejlett funkcionalitást és egyszerű kezelhetőséget biztosít.

5.1. Szimuláció létrehozása

A programozási folyamat első lépéseként kiválasztottam a megfelelő robot modellt, majd beállítottam a preferált kommunikációs módot, amely lehetett USB vagy Ethernet TCP. A programozási nyelvként a legújabb MELFA-Basic VI-t választottam. Mivel a robot nem került lineáris mozgató sínekre, nem volt szükség a Travel axis és additional axis beállításokra.

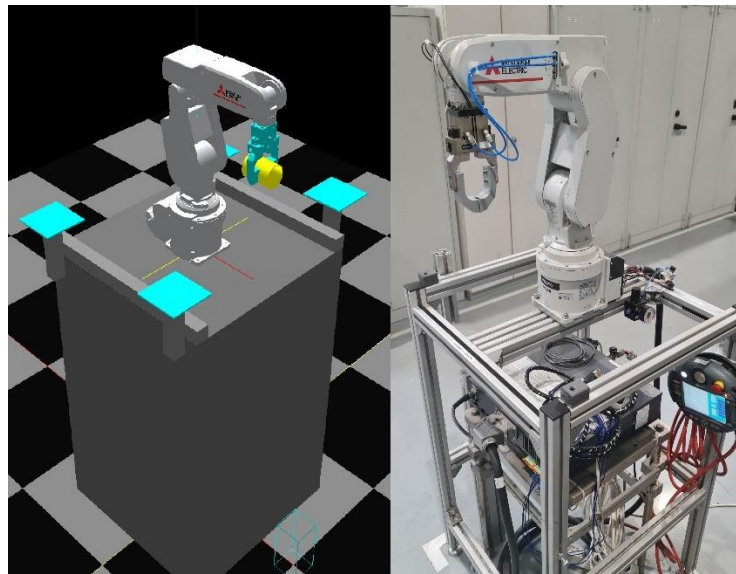
Az RT Toolbox 3 szoftver használatával szimulációkat hoztam létre, amelyek lehetővé tették a robot mozgásainak és műveleteinek előzetes tesztelését. Ez a szoftver segített a robot és a különböző munkaállomások közötti interakciók szimulálásában, biztosítva a hatékonyabb és biztonságosabb munkavégzést.



9. ábra RT Toolbox 3 Project Szerkesztő

5.2. A 3D Monitorban felállított virtuális környezet létrehozása

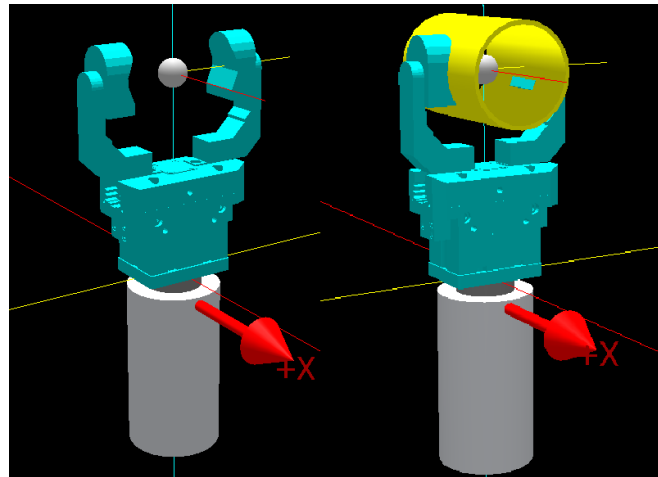
A RT Toolbox 3 segítségével egy virtuális környezetet hoztam létre a 3D Monitorban, amely lehetővé teszi a robot elhelyezkedését és mozgását az általam megírt program alapján. Ebben a virtuális térben a robot képes interakcióba lépni az elhelyezett testekkel. A monitoron csak előre meghatározott formájú idomokat vagy CAD modelleket lehet elhelyezni, ami biztosítja a pontos és hatékony szimulációt.



10. ábra Szimulált és Valós Robot álmás

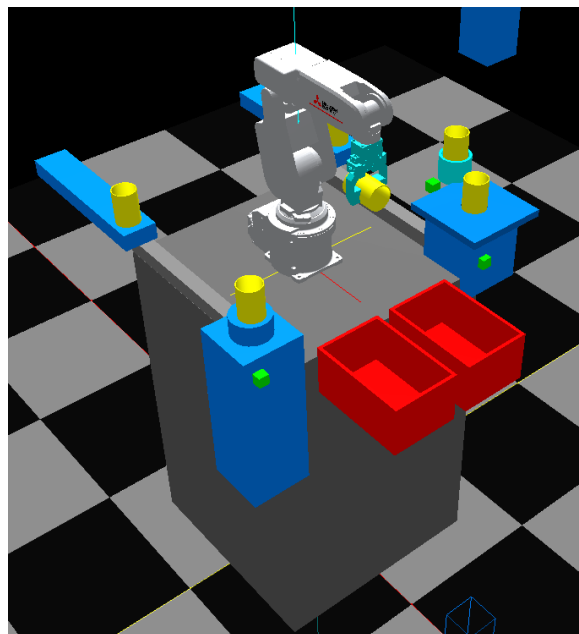
Először elkészítettem a robot állványának modelljét, amelyre a robot fel lesz szerelve. Az állvány modelljét a valóságban létező állvány alapján terveztem meg. A képen látható négy, kékkel megjelölt négyzet jelöli azokat a pozíciókat, ahová a robot képes tárgyakat helyezni és ahol interaktálni tud velük.

A robot megfogó keze két állapotban került modellezésre: nyitott állapotban darab nélkül és zárt állapotban darabbal. Mivel a program nem képes fizikai szimulációkra, nem tudja megjeleníteni a megfogó aktuális állapotát. Ezért minden lehetséges állapotot külön kellett modellezni. A modellek hozzáadása a következőképpen történt: a layout tree-ben a robothoz hozzáadtam az *"add hand"* parancssal. A 11. ábrán látható a megfogók külön állapotai.



11. ábra Hand Modellek

A következő lépésben létrehoztam a szimulált állomásokat, ahova a robotkar a munkadarabokat elhelyezi. A bejövő szalagon érkeznek a nyers darabok. Az első állomás a mágnesezés, ahol a darabot felmágnesezik, és magnetométerrel ellenőrzik. Az állomás oldalán található indikátor jelzi, hogy jó vagy rossz darab készült. A kamera és a mágneses állomás között található piros selejt tároló dobozok.

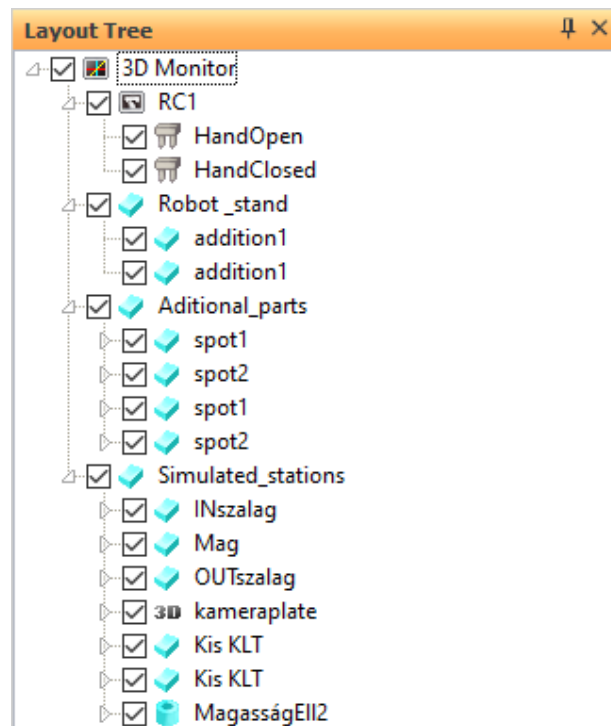


12. ábra Szimulált állomások modellje

A kamerás állomás a következő a folyamatban, ahol a mágnesek pozícióját ellenőrzik. Itt is egy indikátor jelzi a darab megfelelőségét. A következő helyen a magasságmérő helyezkedik el, amely a gyűrűs típusnál ellenőrzi a mágnesek pozícióját. Végül a robot a kimenő szalagra pakolja a darabokat.

A szimulált környezetben az indikátorok három különböző színű, egymásba helyezett kockából állnak. Ezek a kockák az aktuális darab teszten való áthaladásának függvényében változtatják megjelenésüket, amit a program irányít. Az indikátorok színei azonnali vizuális visszajelzést nyújtanak arról, hogy a darab sikeresen teljesítette-e a tesztet.

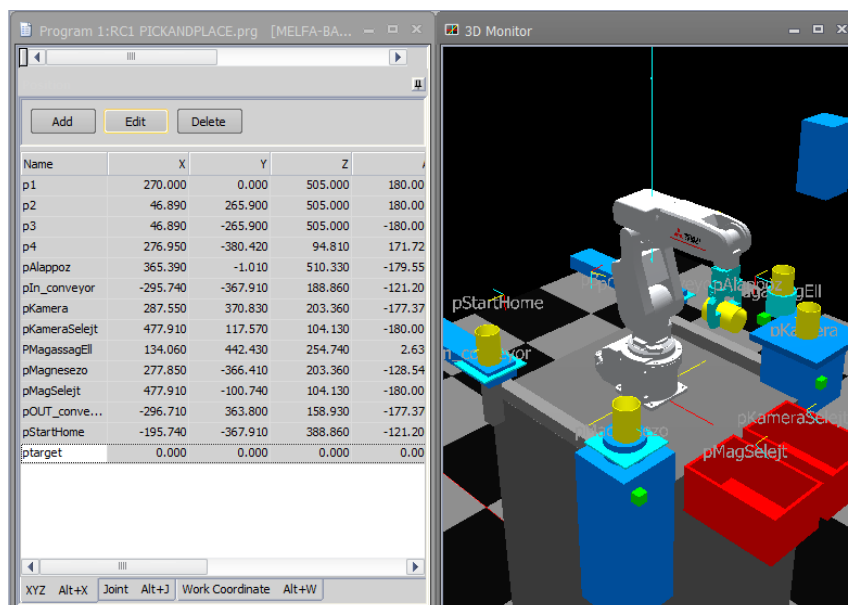
A Layout Tree egy hierarchikus nézet, amely a projektben található összes elemet megjeleníti. Ez magában foglalja a robotokat, a munkadarabokat, az eszközöket és egyéb elemeket, amelyek a szimulációban vagy a valós munkakörnyezetben szerepelnek. A Layout Tree segítségével könnyen áttekinthető a projekt teljes szerkezete, és gyorsan hozzáférhetünk a különböző elemekhez.



13. ábra Layout Tree

5.3. Az útvonalpontok

Az útvonalpontok (waypoint-ok) ebben a szoftverben a robot mozgási útvonalainak előre meghatározott pontjai, amelyek meghatározzák a robot karjának helyzetét és orientációját egy adott időpontban. Ezek a pontok kulcsfontosságúak a pontos és hatékony mozgástervezésben és -végrehajtásban.



14. ábra Robot Program Waypointok

Az XYZ waypoint-ok a robot karjának pozícióját a térbeli koordináta-rendszerben (X, Y, Z) határozzák meg. Ezek a pontok megadják, hogy a robot végső effektorának (például a megfogónak) hol kell lennie a háromdimenziós térben. Az XYZ waypoint-ok különösen hasznosak, ha a robotnak meghatározott helyekre kell eljutnia, például amikor munkadarabokat vesz fel vagy helyez el.

A Joint waypoint-ok a robot karjának egyes ízületeinek (jointjainak) szögértékeit határozzák meg. Ezek az értékek a robot karjának különböző pozícióit és orientációit szabályozzák. A Joint waypoint-ok használatával a robot pontosan irányítható bonyolult mozgások során, mivel minden ízület szöge külön-külön beállítható.

5.3.1. Lineáris és jointos interpoláció a MOV és MVS Utasításokkal

A Mitsubishi RT Toolbox használata során kétféle interpolációt alkalmazok: lineáris és jointos interpolációt, melyek a robot mozgásának pontos irányítását teszik lehetővé a különböző utasítások segítségével.

Lineáris interpoláció (MVS)

A lineáris interpoláció során a robot végpontja (end-effector) egyenes vonalban mozog két pont között. Ez az interpoláció biztosítja, hogy a robot útvonala kiszámítható és közvetlen legyen, ami különösen fontos olyan feladatoknál, ahol a robotnak akadályokat kell elkerülnie vagy pontos pályán kell haladnia. A MVS (Move with Speed) utasítást használom a lineáris interpoláció elérésére. Beállítottam a MVS utasításokat, amikor a robotnak precíziós mozgásokat kellett végeznie, például tárgyak felvétele és

lerakása során. Ez az utasítás biztosítja, hogy a robot végpontja egyenes vonalban haladjon a célpozíció felé, így elkerülve az esetleges akadályokat.

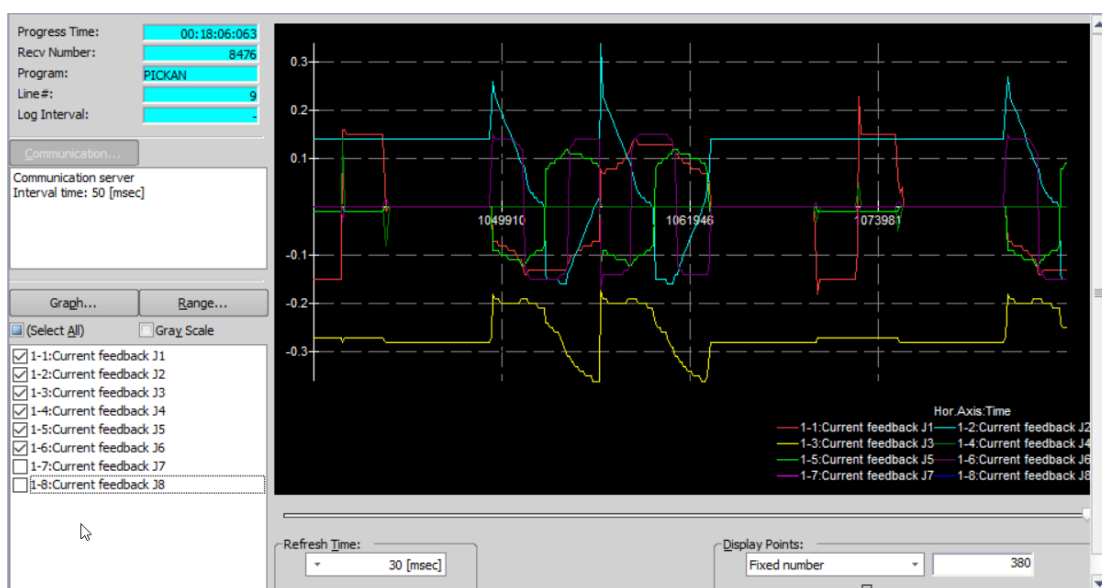
Jointos interpoláció (MOV)

A jointos interpoláció során a robot egyes ízületei (jointjai) szögértékei egyidejűleg változnak, hogy a végpont a kívánt pozícióba kerüljön. Ez a mozgásforma gyakran gyorsabb, mivel a robot ízületei a leghatékonyabb módon mozognak a cél felé, de a mozgás pályája kiszámíthatatlanabb lehet. Az MOV (Movement) utasítást használom a jointos interpoláció elérésére. Ezt az utasítást akkor állítottam be, amikor a robotnak nagyobb távolságokat kellett megtennie, és a mozgás sebessége fontosabb volt, mint a pontos útvonal követése. Az MOV lehetővé teszi, hogy a robot a lehető leggyorsabb úton érje el a célpozíciót, figyelembe véve a megadott sebességet és gyorsulást.

A program hatékonyságának és takarékoságának maximalizálása érdekében fontos, hogy a megfelelő interpolációs módszert és utasítást válasszam a feladat igényei alapján.

A MVS utasítást akkor használom, amikor a mozgásnak pontosnak és akadálymentesnek kell lennie. Ez a módszer biztosítja, hogy a robot előre meghatározott útvonalat követ, ami különösen fontos a precíziós feladatoknál. Az MOV utasítást pedig akkor alkalmazom, amikor a sebesség és a hatékonyság a legfontosabb, és nincs szükség pontos pályakövetésre. Ez a módszer lehetővé teszi a robot számára, hogy a leggyorsabb és legenergiahatékonyabb módon érje el a célját.

Az általam fejlesztett programokban mindkét interpolációs módszert alkalmaztam a feladatok specifikus igényei alapján.



15. ábra Robot mozgások áramfelvétele

A 15. ábrán látható a robot működése során felvett adatok grafikonja. A bal oldalon az MVS utasítások által irányított mozgások láthatók, míg a jobb oldalon a MOV utasításokkal végrehajtott mozgások. Az MVS esetén több motor is egyszerre működik, ami magasabb áramfelvételt eredményez. Ez jól szemlélteti, hogy a jointos interpoláció gyorsabb és hatékonyabb mozgást tesz lehetővé, ugyanakkor nagyobb energiafogyasztással és terheléssel jár a robotkaron.

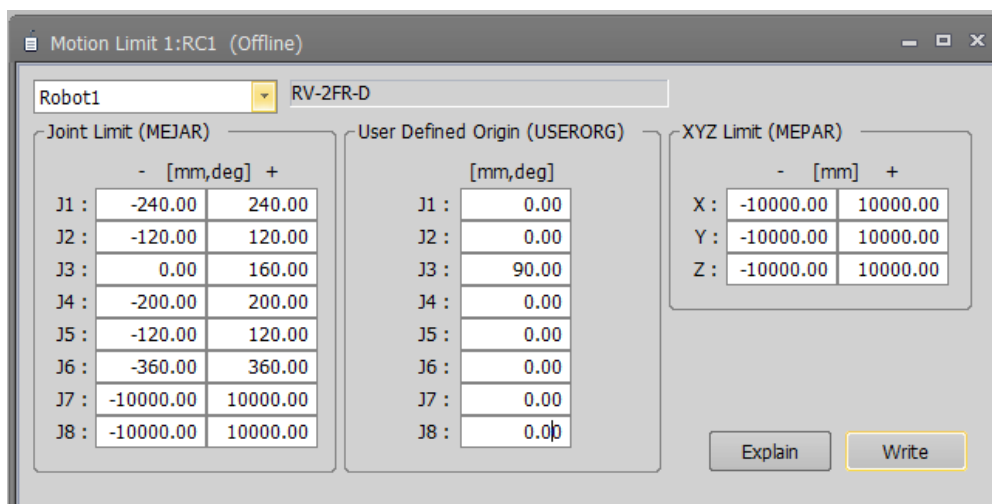
5.4. Robot paraméterek

5.4.1. Mozgási paraméterek

A mozgási paraméterek beállítása alapvető fontosságú a robot pontos és biztonságos működéséhez. A Mitsubishi RT Toolbox-ban számos mozgási paramétert lehet beállítani, amelyek közül néhányat részletesebben ismertetek:

Mozgási határértékek (Motion Limit Parameters)

A mozgási határértékek meghatározzák a robot karjának munkaterületét. Ezek a paraméterek biztosítják, hogy a robot csak a megengedett tartományon belül mozogjon, elkerülve ezzel az esetleges ütközéseket és sérüléseket. A mozgási határértékek beállításához csatlakozni kell a robot vezérlőjéhez, majd a „Movement parameter” menüpont alatt kiválasztani a „Motion Limit” opciót. A paraméterek módosítása után a „Write” gombra kattintva írhatjuk át az értékeket a robot vezérlőjébe.



16. ábra Motion Limit Parameters

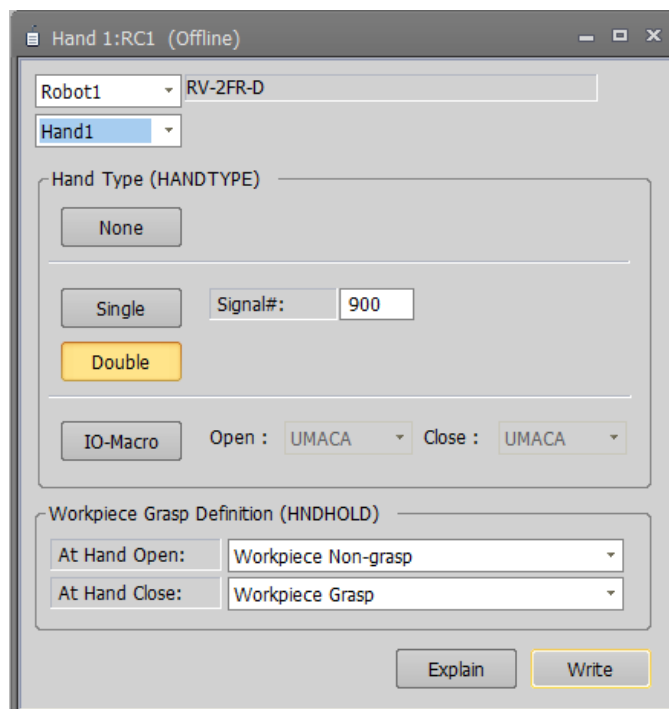
Jog paraméterek (Jog Parameters)

A jog paraméterek a robot egyes ízületeinek (jointjainak) mozgási sebességét határozzák meg. Ezek a paraméterek különösen fontosak a robot manuális mozgása

során, például a pozíció beállításakor. A jog paraméterek beállítása hasonlóan történik a mozgási határértékekhez: csatlakozni kell a robot vezérlőjéhez, majd a „Movement parameter” menüpont alatt kiválasztani a „Jog” opciót. A változtatásokat itt is a „Write” gomb megnyomásával lehet véglegesíteni.

5.4.2. Kéz paraméterek (Hand Parameters)

A kéz paraméterek beállítása lehetővé teszi a robot megfogójának (gripper) típusának és működésének meghatározását. A kéz paraméterek beállítása során megadhatjuk, hogy a megfogó egy vagy két szolenoidot használ-e, valamint, hogy nyitott vagy zárt állapotban van-e az alapértelmezett pozícióban. A paraméterek beállításához a „Movement parameter” menüpont alatt a „Hand” opciót kell kiválasztani, majd a megfelelő értékek megadása után a „Write” gombbal írhatjuk át azokat a robot vezérlőjébe.



17. ábra Hand Parameters

5.4.3. Súly és méret paraméterek (Weight and Size Parameters)

A súly és méret paraméterek segítségével meghatározhatjuk a robot által kezelt tárgyak maximális tömegét és méreteit. Ezek a paraméterek fontosak a robot stabilitásának és biztonságos működésének biztosításához. A megfelelő paraméterek beállításával elkerülhetjük a túlterhelést és az ebből adódó meghibásodásokat.

			WRKDAT0	WRKDAT1	WRKDAT2	WRKDAT3	WRKDAT4	WRKDAT5	WRKDAT6	WRKDAT7	WRKDAT8
Robot1			RV-2FR-D								
Work	Weight [Kg] :		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Size [mm]	X :	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Y :	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Z :	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Center of Gravity Position [mm]	X :	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Y :	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Z :		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
			HNDDAT0	HNDDAT1	HNDDAT2	HNDDAT3	HNDDAT4	HNDDAT5	HNDDAT6	HNDDAT7	HNDDAT8
Hand	Weight [Kg] :		3.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
	Size [mm]	X :	200.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Y :	200.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Z :	150.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Center of Gravity Position [mm]	X :	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		Y :	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Z :		100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	

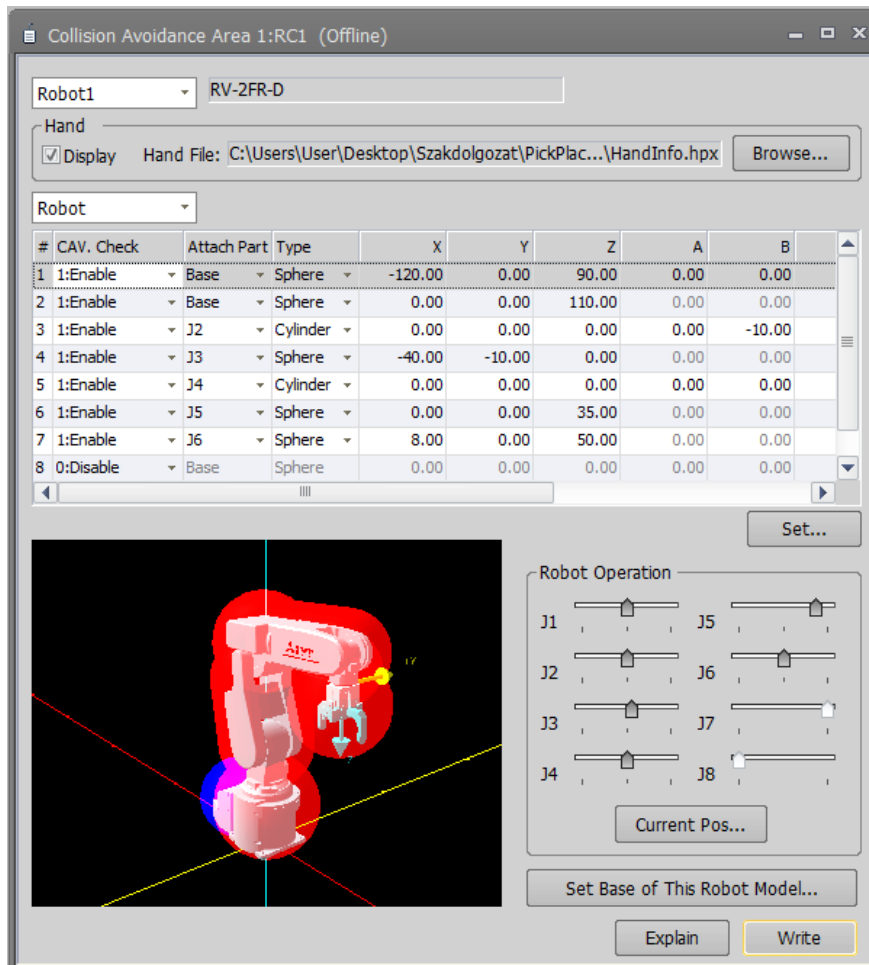
18. ábra Súly- és méretparaméterek

5.4.4. Eszköz paraméterek (Tool Parameters)

Az eszköz paraméterek beállítása során megadhatjuk a robot által használt szerszám típusát és annak jellemzőit. Ez magában foglalja a szerszám tömegét, méreteit és az alkalmazott erőket. Az eszköz paraméterek pontos beállítása biztosítja a robot számára, hogy optimális módon tudjon dolgozni a különböző feladatok során.

5.4.5. Ütközés-észlelési paraméterek (Collision Detection Parameters)

Az ütközés-észlelési paraméterek beállításával a robot képes lesz felismerni és elkerülni az esetleges ütközéseket a munkaterületén belül. Ezek a paraméterek különösen fontosak a biztonságos működés szempontjából, mivel lehetővé teszik a robot számára, hogy időben reagáljon a váratlan akadályokra.



19. ábra Ütközés elkerülés Paraméter

Az ütközésselkerülési zónák meghatározása és beállítása lehetővé teszi a robot számára, hogy érzékelje és elkerülje a potenciálisan veszélyes területeket. Ezek a zónák virtuális határokat jelölnek ki a robot munkaterületén belül, amelyeken belépve a robot automatikusan lassít vagy megáll, hogy elkerülje az ütközést.

Először definiálni kell a zónák határait koordináták vagy a robot mozgási tartományai alapján. A zónák paramétereinek beállítása során megadhatók a zónák méretei és pozíciói, valamint a hozzájuk tartozó reakciók, mint például a lassítási sebesség és a megállási távolság.

5.5. A robot program

A robotcella programját négy fő részre osztottam:

- Robot Vezérlő Program: Ez a fő program felelős a robot irányításáért, ellátja a robot alapvető feladatait, és kommunikál a különböző állomásokkal. A vezérlő program biztosítja, hogy a robot pontosan és hatékonyan végezze el a feladatokat a megadott munkafolyamatok szerint.

- **Mágneses Állomás Programja:** Ehhez a részhez külön programot hoztam létre, amely a mágneses állomás működését szimulálja. Ez a program biztosítja, hogy a mágneses folyamat megfelelően működjön, és az állomás pontosan végezze el a mágneses feladatokat.
- **Kamerás Állomás Programja:** A kamerás állomás működéséhez szintén külön program készült. Ez a program szimulálja a kamerás állomás funkcióit, ellenőrzi a munkadarabokat, és biztosítja a szükséges minőségellenőrzést.
- **Bejövő Anyagáramlás Szimulációja:** Az utolsó program a bejövő anyagáramlást szimulálja. Ez a rész biztosítja, hogy a nyersanyagok megfelelően kerüljenek be a rendszerbe, és a robot számára elérhetőek legyenek a feldolgozáshoz.

Ezek a programrészek együttműködve biztosítják a robotcella hatékony és pontos működését, szimulálva a valós gyártási folyamatokat egy laboratóriumi környezetben.

5.5.1. Robot vezérlő program

A robotot vezérlő program előkészítése (Before_Start) az indulást megelőző folyamatokat foglalja magában. Ez a szakasz kulcsfontosságú a robot megfelelő működéséhez, mivel itt történik az I/O lista beállítása, a szükséges lépések elvégzése, valamint a változók deklarálása.

Ellenőrizni kell, hogy a szervomotor be van-e kapcsolva. Ha nincs, a program bekapcsolja a szervomotort, hogy a robot mozgatható legyen.

```
1 '-----Before_Start-----
2 If M_Svo = 0 Then Servo On      'Ha szervo off akkor kapcsold be
```

Illetve A program futásának kezdetén beállítják a robot mozgási sebességét és az adatkommunikációs protokollokat.

```
3 OvrD 100 'A sebesség 100%-ra állítása
4 OAd1 On
```

A program egyik első lépése a robot I/O portjainak konfigurálása, amely magában foglalja a bemeneti és kimeneti portok hozzárendelését a különböző szenzorokhoz, aktuátorokhoz és más perifériákhoz. A mágneses állomás és a kamerás állomás vezérlőjelek hozzárendelése az adott portokhoz.

```
5 '-----Mágnesező I/O-----
6 MMagSwitch      = 2019      'Mágnesező indító gombja
7 MMagBusy        = 2020      'Mágnesező elfoglalt
8 MMagReady       = 2021      'Mágnesező ready
9 MMagOK          = 2022      'Mágneseség teszten átment a darab
10 MMagNOK        = 2023      'Mágneseség teszten megbukott a darab
11 '-----Kamera I/O-----
```

```

12 MKameraSwitch      = 2029      'Kamerázás indító gombja
13 MKameraBusy        = 2030      'Kamera elfoglalt
14 MKameraReady       = 2031      'Kamera ready
15 MKameraOK          = 2032      'Kamera teszten átment a darab
16 MKameraNOK         = 2033      'Kamera teszten megbukott a darab
17 '-----MagassagEll I/O-----
18 MMagasagSwitch     = 2034      'Magassag ellenörzés indító gombja
19 MMagasagOK         = 2035      'Magasság Ellenörzésen átment
20 MMagasagNOK        = 2036      'Magasság Ellenörzésen megbukott
21 MGyurusTipus       = 2037      'Gyürs típus Input
22 '-----PartSimulation-----
23 MPart_In_conveyor  = 2040
24 MPart_Magnesezo    = 2041
25 MPart_Kamera       = 2042
26 MPart_Out_conveyor = 2043
27 MPart_MaggasagEll = 2044

```

A programban szükséges a változók deklarálása és inicializálása. Ezek a változók tárolják az állapotinformációkat, számlálókat és egyéb adatokat, amelyekre a program futása során szükség van.

```

28 '-----Változók-----
29 Def Inte MMagNOKDB      'Mágnesező selejt darab számláló
30 MMagNOKDB%             = 0
31 Def Inte MKameraNOKDB   'Kamera selejt darab számláló
32 MKameraNOKDB%          = 0
33 Def Inte MOKDB         'Jó darab számláló
34 MOKDB%                 = 0
35 Def Inte MState        'State változó a Case funkcióhoz
36 MState%                 = 0

```

Ezekután a program a fő ciklusba vált, ahol a (Select Case) parancs segítségével meghatározható, hogy a robot melyik műveletet hajtsa végre a program futása során. Ez a módszer lehetővé teszi a kód átláthatóságát és karbantarthatóságát. A Select Case parancs az MState% változó értékét vizsgálja, amely meghatározza, hogy a program melyik szakaszát kell végrehajtani. Az egyes ágak (esetek) különböző értékekhez tartoznak, és mindegyik ág egy adott műveletet vagy műveletsorozatot tartalmaz. Az ágak Case kulcsszóval kezdődnek és az adott értéket követően a végrehajtandó kód következik. A program addig fut amíg a 'State' változó 10 alatt van ez a folyamat gyors leállítását segíti.

```

37 '-----Start-----
38 While MState% < 10      'A főprogram addig fut amíg a While igaz
39 Select MState%

```

Case 0: Az alaphelyzetbe állítás szakasza. Ez a rész reseteli az I/O jeleket, beállítja a robot pozícióját és ellenőrzi a selejtszámlálót. Ha a selejtek száma meghaladja a 10-et, a program az állapotváltozót 10-re állítja, ami leállítja a programot. Ha nem, akkor az állapotváltozót 1-re állítja, hogy a következő szakaszra lépjen.


```

40 '-----Alaphelyzet-----
41 Case 0 'Case 0 : Alaphelyzet a Simulált darabok
alaphelyzetbe állítása
42 M_Out(MPart_Magnesezo) =0 ' és a Ok Nok Jelek resetelése.
43 M_Out(MPart_Kamera) =0
44 M_Out(MPart_Out_conveyor) =0
45 M_Out(MMagOK) =0
46 M_Out(MMagNOK) =0
47 M_Out(MKameraOK) =0
48 M_Out(MKameraNOK) =0
49 M_Out(MMagasagOK) =0
50 M_Out(MMagasagNOK) =0
51 M_Out(MGyurusTipus) =1

```

A robot aktuális pozíciójának mentése és 10 mm-es lépésekben történő felemelése 350 mm magasságig, hogy biztonságos távolságra legyen a következő mozgás előtt.

```

53 ptarget = P_Curr 'ptarget a jelenlegi pozícióra állítása
54 While P_Curr.Z < 350 'ptarget 10mm feljebb tolása addig
55 ptarget.Z = ptarget.Z +10 'amíg el nem érjük a 350mm magasságot
56 Mvs ptarget
57 WEnd
58 Cnt 1 ,200
59 Mov pAlappoz
60 Cnt 0

```

Ellenőrzésre kerül, hogy az összes selejt száma meghaladja-e a 10-et. Ha igen, a program leáll és resetelésre kerül. A robot megfogójának kinyitása az alaphelyzetbe állítás részeként történik, hogy készen álljon az új munkadarabok felvételére. Az állapot változó frissítése, hogy a program a következő lépésre lépjen 'MState% = 1'.

```

61 '62 If (MMagNOKDB% + MKameraNOKDB%) >= 10 Then 'Ha az összes selejt száma
nagyobb mint 10 stop and reset
63 MState% = 10
64 EndIf
65 HOpen 1
66 MState% = 1
67 Break

```

Az alábbi részben bemutatott kód a különböző állomások kezelését végzi. Minden egyes Case ág egy alprogramot hív meg, amely egy adott állomást kezel:

- Case 1: Bejövő szalag kezelése
- Case 2: Mágnesező állomás kezelése
- Case 3: Kamerás állomás kezelése
- Case 4: Magasság ellenőrző állomás kezelése
- Case 5: Kimenő szalag kezelése

Ez a struktúra biztosítja, hogy a program logikusan és szervezeten hajtsa végre a különböző feladatokat, és minden egyes állomás megfelelően működjön a robotcella teljes folyamatában. Az állapotok közötti váltás az MState% változó értékének megfelelően történik, amely lehetővé teszi a folyamatok szekvenciális végrehajtását.

```

69 Case 1
70   GoSub *In_szallag           'Bejövő szalagot kezelő subprogram meghívás
71 Case 2
72   GoSub *Magnesezó          'Mágnesezőt kezelő subprogram meghívás
73 Case 3
74   GoSub *Kamera             'Kamerát kezelő subprogram meghívás
75 Case 4
76   GoSub *Magassag_Ell       'Magasság ellenőrzőt kezelő subprogram meghívás
77 Case 5
78   GoSub *Out_szallag        'Kimenő szalagot kezelő subprogram meghívás
79   MState% = 0
80 Break
81 End Select
82 WEnd

```

Az alábbi kódrészlet bemutatja a bejövő szalag kezeléséért felelős alprogram működését. Ez az alprogram a beérkező munkadarabok felvételét és a megfelelő pozícióba helyezését végzi el, hogy a további feldolgozási lépések zökkenőmentesen folytathassanak.

```

84 '-----In_szallag(1)-----
85 *In_szallag
86 Mov pIn_conveyor +(100.00,+0.00,+200.00) 'Szalag megközelít
87 Wait M_Out(MPart_In_conveyor)=1         'Darabra vár
88 Mov pIn_conveyor +(0.00,+0.00,+100.00)  'Szalag felett
89 Mvs pIn_conveyor                         'Szalag lennt
90 Dly 0.5
91 HClose 1
92 M_Out(MPart_In_conveyor) =0
93 Mvs pIn_conveyor +(0.00,+0.00,+100.00)   'Szalag felett
94 Dly 1
95 MState% = 2
96 Return

```

Először a robot a bejövő szalaghoz közelít, és várja a munkadarab érkezését. Amikor a darab megérkezik, a robot pozicionálja magát fölötte, leereszkedik, majd a megfogójával felveszi a darabot. A Szimulációs jelzés nullázása után a robot visszaemelkedik, és készen áll a következő műveletre. Végül az állapotváltozót beállítja, hogy a program a következő lépésre léphessen.

Az alábbi kódrészlet a mágnesező állomás kezeléséért felelős alprogramot mutatja be, amely a munkadarabok mágnesezését és a folyamat ellenőrzését végzi.

```

97 '-----Magnesező(2)-----
98 *Magnesezó
99 Wait M_Out(MMagReady) = 1             'Mágnesezó Kész Jelre vár
100 Mov pMagnesezó + (0.00,+0.00,+100.00) 'Mágnesező fölött
101 Mvs pMagnesezó                       'Mágnesezőben
102 HOpen 1
103 M_Out(MPart_Magnesezó) =1
104 Mvs pMagnesezó + (0.00,+0.00,+100.00) 'Mágnesező fölött
105 M_Out(MMagSwitch)= 1                 'Mágnesező Start Jel
106 Dly 1
107 Wait M_Out(MMagReady) = 1             'Mágnesező Kész Jel
108 Mvs pMagnesezó                       'Mágnesezőben
109 HClose 1
110 M_Out(MPart_Magnesezó) =0
111 Mvs pMagnesezó + (0.00,+0.00,+100.00)
112 If M_Out(MMagOK)=0 And M_Out(MMagNOK)=1 Then'Selejtezés

```

```

113   Mov pMagSelejt
114   HOpen 1
115   MMagNOKDB% = MMagNOKDB% + 1
116   Dly 1
117   Mov pMagSelejt +(0.00,0.00,+250.00)
118   MState% =0
119   ElseIf M_Out(MGyurusTipus)=0 Then           'Nem gyürüs típus
120     MState% = 3
121   ElseIf M_Out(MGyurusTipus)=1 Then           'Gyürüs típus
122     MState% = 4
123   EndIf
124   Return

```

Az alprogram először a mágnesező kész jelre vár, majd a robot a mágnesező fölé mozog és leereszkedik. A megfogó kinyitása után a mágnesező indítására jelzést küld, majd a mágnesezés befejezésére vár. Ezután a robot visszazárja a megfogót, és ellenőrzi a mágnesezés eredményét. Ha a folyamat nem volt sikeres, a munkadarabot a selejtező pozícióba helyezi. Ha a munkadarab mágnesezése sikeres, a robot ellenőrzi a gyűrüs típust. Nem gyűrüs típus esetén a program a következő állomásra (kamera) lép, gyűrüs típus esetén pedig a magasság ellenőrző állomásra. Végül az állapotváltozót beállítja a következő művelethez.

Az alábbi kódrészlet a kamerás állomás kezeléséért felelős alprogramot mutatja be, amely a munkadarabok vizuális ellenőrzését végzi.

```

125 '-----Kamera(3)-----
126 *Kamera
127 Wait M_Out(MKameraReady) = 1           'Kamera Kész Jelre vár
128 Mov pKamera + (+0.00,+0.00,+100.00)     'Kamera felett
129 Mvs pKamera                             'Kamerában
130 HOpen 1                                 'Darabot lerak
131 M_Out(MPart_Kamera) =1
132 Mvs pKamera + (+0.00,+0.00,+100.00)     'Kamera felett
133 Mov pKamera + (+0.00,-100.00,+100.00)   'Kamera előtt
134 M_Out(MKameraSwitch)= 1
135 Dly 1
136 Wait M_Out(MKameraReady) = 1
137 Mov pKamera + (+0.00,+0.00,+100.00)     'Kamera felett
138 Mvs pKamera                             'Kamerában
139 HClose 1                                'Darabot felvesz
140 M_Out(MPart_Kamera) =0
141 Mvs pKamera + (+0.00,+0.00,+100.00)
142 If M_Out(MKameraOK) = 0 And M_Out(MKameraNOK) = 1 Then 'Selejtezés
143   Mov pKameraSelejt
144   HOpen 1
145   MKameraNOKDB% = MKameraNOKDB% + 1
146   Dly 1
147   Mov pKameraSelejt +(0.00,+0.00,+250.00)
148   MState% = 0
149 Else
150   MState% = 5
151 EndIf
152 Return

```

Az alprogram először a kamera kész jelre vár, majd a robot a kamera fölé mozog és leereszkedik. A megfogó kinyitása után a robot elhelyezi a munkadarabot a kamera pozíciójában, majd a kamera indítására jelzést küld. A kamera vizsgálata után a robot

felveszi a munkadarabot, és ellenőrzi a vizsgálat eredményét. Ha a vizsgálat nem sikerült, a munkadarabot a selejtező pozícióba helyezi, különben a következő állomásra lép. Végül az állapotváltozót beállítja a következő művelethez.

Az alábbi kódrészlet a magasság ellenőrző állomás kezeléséért felelős alprogramot mutatja be, amely a munkadarabok magasságának ellenőrzését végzi.

```
153 '-----Magassag_Ell(4)-----
154 *Magassag_Ell
155 Mov PMagassagEll +(+0.00,-100.00,+100.00)
156 Mov PMagassagEll +(+0.00,+0.00,+100.00)
157 Mvs PMagassagEll
158 HOpen 1 'Darabot lerak
159 M_Out(MPart_MaggasagEll) =1
160 Dly 1
161 M_Out(MMagasagSwitch) = 1
162 If (Rnd (0)*100) < 80 Then 'Rand operator eldönti hogy Ok vagy Nok 88%
arányal
163 M_Out(MMagasagOK) = 1
164 M_Out(MMagasagNOK) = 0
165 Else
166 M_Out(MMagasagOK) = 0
167 M_Out(MMagasagNOK) = 1
168 EndIf
169 M_Out(MMagasagSwitch) = 0
170 Dly 1
171 HClose 1 'Darabot felvesz
172 M_Out(MPart_MaggasagEll) =0
173 Mvs PMagassagEll +(+0.00,+0.00,+100.00)
174 If M_Out(MMagasagOK) = 1 And M_Out(MMagasagNOK) = 0 Then 'Selejtezés
175 MState% = 5
176 Else
177 Mov pKameraSelejt +(+0.00,+0.00,+250.00)
178 Mov pKameraSelejt
179 HOpen 1
180 MKameraNOKDB% = MKameraNOKDB% + 1
181 Dly 1
182 Mov pKameraSelejt +(+0.00,+0.00,+250.00)
183 MState% = 0
184 EndIf
185 Return
```

Az alprogram először a magasság ellenőrző állomás fölé mozog és leereszkedik. A megfogó kinyitása után a munkadarabot a magasság ellenőrző állomásra helyezi, majd elindítja az ellenőrzést. A véletlenszerű döntés alapján meghatározza, hogy a munkadarab megfelel-e. Ha igen, a következő állomásra lép. Ha nem, a munkadarabot a selejtező pozícióba helyezi. Végül az állapotváltozót frissíti, hogy a program a következő műveletre léphessen.

Az alábbi kódrészlet a kimenő szalag kezeléséért felelős alprogramot mutatja be, amely a kész munkadarabokat a kimenő szalagra helyezi.

```
186 '-----Out_szallag(5)-----
187 *Out_szallag
188 Mov pOUT_conveyor
189 HOpen 1
190 MOKDB% = MOKDB% + 1
191 M_Out(MPart_Out_conveyor) =1
192 Dly 0.5
193 Mvs pOUT_conveyor + (+0.00,+0.00,+100.00)
194 Dly 1
195 Return
```

Az alprogram először a kimenő szalag pozíciójához mozog, majd a megfogó kinyitásával elengedi a munkadarabot. Növeli a jó darabok számlálóját, és jelzést küld a kimenő szalagnak. Rövid várakozás után a robot visszaemelkedik a kimenő szalag fölé, majd visszatér a főprogramhoz, és ezzel bezárul a ciklus.

5.5.2. Mágnesező állomás program

Az alprogram először vár a mágnesező start jelére (M_Out(MMagSwitch) = 1). Amikor a jel megérkezik, a mágnesező állomást foglalként (M_Out(MMagBusy) = 1) és nem készként (M_Out(MMagReady) = 0) állítja be. Ezután reseteli a szükséges változókat és időzítőt.

A While ciklusban az időzítő segítségével figyeli az eltelt időt. Amikor az időzítő eléri a beállított mágnesezési időt (MMagTime), a program randomizálja a mágnesezés eredményét. Ha a random érték kisebb, mint a határérték (MMagLimit), a mágnesezés sikeres (M_Out(MMagOK) = 1), ellenkező esetben sikertelen (M_Out(MMagNOK) = 1).

A ciklus végén a program visszaállítja a mágnesező állomást üres és üzemképes állapotba (M_Out(MMagBusy) = 0, M_Out(MMagReady) = 1), kikapcsolja a start jelzést (M_Out(MMagSwitch) = 0), és reseteli az időzítőt.

```
16 M_Out(MMagSwitch) = 0
17 '-----
18 Wait M_Out(MMagSwitch) = 1
19 '
20 M_Out(MMagBusy) = 1
21 M_Out(MMagReady) = 0
22 '
23 'While (M_Out(MMagReady) = 0)'And (M_Out(MMagSwitch)% = 1) 'Még egy while
szükséges a ciklus elfő felét csak egyszer futayuk ll
24 'Használt változók resetelése
25 '
26 M_Timer(2)=0
27 M_Out(MMagOK) = 0
28 M_Out(MMagNOK) = 0
29 '
30 While (M_Out(MMagReady) = 0) 'And (M_Out(MMagSwitch)% = 1) 'While loop az idő
kiszámítására.
31 'Mtime = M_Timer(2)
32 '
```

```

33     If M_Timer(2) > MMagTime*1000 Then
34         'Értékek eandomizálása simulációban.
35         MRand% = 100*Rnd(0)
36         If MRand% < MMagLimit Then
37             M_Out(MMagOK) =1
38             M_Out(MMagNOK) =0
39         Else
40             M_Out(MMagOK) =0
41             M_Out(MMagNOK) =1
42         EndIf
43         'A gép újra üres és üzemképes
44         M_Out(MMagBusy) = 0
45         M_Out(MMagReady) = 1
46         M_Out(MMagSwitch) = 0
47         M_Timer(2) = 0
48     EndIf
49 '
50 WEnd

```

A mágneses állomás programja a munkadarabok hatékony és pontos mágnesezését végzi. A randomizálási mechanizmus szimulálja a valós mágnesezési folyamatot, amelynek során az állomás jelzi a sikeres vagy sikertelen műveleteket. Az állomás a folyamat végén készen áll a következő darab feldolgozására, biztosítva a folyamat zökkenőmentes folytatását.

5.5.3. Kamera állomás program

A kamerás állomás programja a munkadarabok vizuális ellenőrzését végzi, amely egy kritikus lépés a gyártási folyamatban. Ez a program biztosítja, hogy minden egyes munkadarab megfelel az előírt minőségi követelményeknek, mielőtt továbbhaladna a következő gyártási lépésre.

A program a munkadarabok vizsgálatát egy szimulált környezetben végzi, ahol a kamerás ellenőrzés eredményeit egy randomizálási mechanizmus határozza meg. Ez a mechanizmus a valós kamerázási folyamatot szimulálja, amely során az állomás véletlenszerűen határozza meg, hogy a munkadarab megfelelt-e a teszten vagy sem. A randomizálás azért szükséges, hogy a szimuláció során a valós környezethez hasonlóan változatos eredményeket kapjunk.

```

1 M_Timer(3)=0 'A 2. Timert használjuk mert az 1 már használja a robot a belső tack
time kiszámolásásra
2 Mtime = 0
3 '
4 MKameraSwitch = 2029 'Kamerázás indító gombja
5 MKameraBusy = 2030 'Kamera elfoglalt
6 MKameraReady = 2031 'Kamera ready
7 MKameraOK = 2032 'Kamera teszten átment a darab
8 MKameraNOK = 2033 'Kamera teszten megbukott a darab
9 MKameraLimit = 80 'Kamerázás szimulált selejtezés limitje (%)
10 MKameraTime = 2 'Kamerázás szimulált idő tartama (sec)
11 '
12 M_Out(MKameraBusy) = 0 'MKameraBusy Gép foglalt
13 M_Out(MKameraReady) = 1 'MKameraReady Gép Ready
14 'M_Out(MKameraOK) = 0 'MKameraOK OK Darab

```

```

15 'M_Out(MKameraNOK) = 0      'MKameraNOK NOK Darab
16 '
17 Def Inte MRand
18 M_Out(MKameraSwitch) = 0
19 '-----
20 Wait M_Out(MKameraSwitch) = 1
21 '
22 M_Out(MKameraBusy) = 1
23 M_Out(MKameraReady) = 0
24 '
25 'While (M_Out(MKameraReady) = 0)'And (M_Out(MKameraSwitch)% = 1) 'Még egy while
szükséges a ciklus elfő felét csak egyszer futayuk ll
26 'Használt változók resetelése
27 '
28     M_Timer(3)=0
29     M_Out(MKameraOK) = 0
30     M_Out(MKameraNOK) = 0
31 '
32     While (M_Out(MKameraReady) = 0) 'And (M_Out(MKameraSwitch)% = 1) 'While loop
az idő kiszámítására.
33     'Mtime = M_Timer(3)
34     If M_Timer(3) > MKameraTime*1000 Then
35         'Értékek eandomizálása simulációban.
36         MRand% = 100*Rnd(0)
37         If MRand% < MKameraLimit Then
38             M_Out(MKameraOK) =1
39             M_Out(MKameraNOK) =0
40         Else
41             M_Out(MKameraOK) =0
42             M_Out(MKameraNOK) =1
43         EndIf
44         'A gép újra üres és üzemképes
45         M_Out(MKameraBusy) = 0
46         M_Out(MKameraReady) = 1
47         M_Out(MKameraSwitch) = 0
48         M_Timer(3) = 0
49     EndIf
50 '
51 '
52 WEnd

```

A program elején a harmadik időzítőt használjuk, mivel az első már foglalt. Beállítjuk a kamerázás indítógombját (MKameraSwitch), a kamera elfoglaltságát (MKameraBusy), készenléti állapotát (MKameraReady), a teszt eredményeit (MKameraOK és MKameraNOK), a selejtezési limitet (MKameraLimit), és a szimulált kamerázási időtartamot (MKameraTime).

A kamera alaphelyzetben nem foglalt ($M_Out(MKameraBusy) = 0$) és készen áll ($M_Out(MKameraReady) = 1$). A program indulásakor a MKameraSwitch nullázva van. A program vár a kamera indító jelére. Amikor a jel megérkezik, a kamera elfoglalt állapotba kerül ($M_Out(MKameraBusy) = 1$), és nem készenléti állapotba ($M_Out(MKameraReady) = 0$). A szükséges változókat és az időzítőt reseteljük.

A While ciklus során figyeljük az eltelt időt. Amikor az időzítő eléri a beállított kamerázási időt (MKameraTime), a program randomizálja a teszt eredményét. Ha a random érték kisebb, mint a limit (MKameraLimit), a teszt sikeres ($M_Out(MKameraOK) = 1$), egyébként sikertelen ($M_Out(MKameraNOK) = 1$).

A ciklus végén a kamera állomás újra üres és üzemképes lesz ($M_Out(MKameraBusy) = 0$, $M_Out(MKameraReady) = 1$), a start jelzés kikapcsol ($M_Out(MKameraSwitch) = 0$), és az időzítőt reseteljük.

5.5.4. Bejövő anyagáramlás szimulációja

A bejövő szalag programja hatékonyan szimulálja a munkadarabok érkezését a gyártósorra. A program egy véletlenszerű időintervallumot generál, amely után a szalag érkezési jelzését bekapcsolja. Ez a mechanizmus biztosítja, hogy a szalag nem folyamatosan működik, hanem véletlenszerű időközönként, ami jobban tükrözi a valós gyártási környezetet. Ezzel a módszerrel a gyártósor hatékonyan és realiztikusán kezeli a beérkező munkadarabokat, hozzájárulva a folyamat zökkenőmentes és hatékony működéséhez.

```
1 Def Inte MNextWorkpeace
2 MPart_In_conveyor = 2040
3 '
4 MNextWorkpeace% = 5+10*Rnd(0)
5 M_Timer(4)=0
6 '
7 Wait M_Timer(4) > MNextWorkpeace%*1000
8 M_Out(MPart_In_conveyor) = 1
9 '
```

Először definiálom a MNextWorkpeace változót, majd beállítom a bejövő szalag állapotát a MPart_In_conveyor változóval (2040). Ezután egy véletlenszerű várakozási időt generálok, amely 5 és 15 másodperc között mozog. Nullázom az időzítőt, majd várok, amíg az időzítő eléri a generált várakozási időt. Amikor az idő letelik, bekapcsolom a bejövő szalag jelzését, jelezve, hogy egy új munkadarab érkezett.

A program véletlenszerű időközönként szimulálja a munkadarabok érkezését a szalagon, ami realiztikusabbá teszi a szimulációt és biztosítja a folyamatos működést.

6. Összegzés

A dolgozatom célja az ipari robotok tervezésének és alkalmazásának bemutatása laboratóriumi környezetben, különös tekintettel a pick and place feladatokra. Az ipari automatizálás terén a robotok egyre nagyobb szerepet kapnak, mivel képesek komplex feladatokat precízen és gyorsan elvégezni, ezzel növelve a termelékenységet és csökkentve a költségeket. A dolgozat során részletesen ismertettem a technológia fejlődését, a fejlesztési eszközök kiválasztásának folyamatát, valamint a robotvezérlés megvalósításának lépéseit és annak tesztelését.

Az első fejezetben áttekintettem a robotika alapjait és az ipari automatizálás jelenlegi helyzetét. A második fejezetben a robotok jövőjét és egy kiválasztási példát mutattam be. Részletesen tárgyaltam az ipari robotok fejlődési trendjeit és bemutattam egy példát, amely során összehasonlítottam különböző robotokat, és végül a legmegfelelőbbet választottam ki a feladatra.

A harmadik fejezetben a laboratóriumban használt eszközöket és perifériákat ismertettem. Bemutattam a laborban alkalmazott különböző robotokat és azok programozási lehetőségeit, valamint a használt kiegészítő eszközöket és perifériákat, amelyek elengedhetetlenek a hatékony robotvezérléshez.

A negyedik fejezet a megvalósítandó feladattal, a pólusház ellenőrző állomással foglalkozott. Bemutattam a robotcella tervezésének és megvalósításának lépéseit, valamint részletesen ismertettem a feladat specifikus követelményeit és a megvalósítás során szerzett tapasztalatokat. Az ellenőrző állomás működésének lépéseit is bemutattam.

Az ötödik fejezet a robotprogramozás folyamatát tárgyalta. Részletesen ismertettem a programozási lépéseket, beleértve a különböző interpolációs módszerek (lineáris és jointos) alkalmazását a robot mozgásának pontos irányítására. Bemutattam a programozáshoz használt szoftvereket és azok funkcióit, valamint a programozás során felmerült kihívásokat és azok megoldásait.

Összegzésként elmondható, hogy a dolgozatban bemutatott megoldások és fejlesztési lépések hozzájárulnak az ipari automatizálás hatékonyságának növeléséhez, valamint a gyártási folyamatok optimalizálásához. A tesztelések során szerzett tapasztalatok és a felmerült kihívások sikeres kezelése lehetővé teszi a robotrendszerek további fejlesztését és alkalmazását a jövőben.

7. Summary

The aim of my thesis is to present the design and application of industrial robots in a laboratory environment, with a particular focus on pick and place tasks. In the field of industrial automation, robots play an increasingly significant role as they are capable of performing complex tasks precisely and quickly, thereby increasing productivity and reducing costs. Throughout the thesis, I detailed the development of the technology, the process of selecting development tools, as well as the steps involved in implementing and testing robot control systems.

In the first chapter, I reviewed the basics of robotics and the current state of industrial automation. The second chapter discussed the future of robots and provided an example of robot selection. I elaborated on the development trends of industrial robots and presented a selection example where I compared different robots and ultimately chose the most suitable one for the task.

The third chapter focused on the tools and peripherals used in the laboratory. I introduced the various robots used in the lab and their programming capabilities, as well as the supplementary tools and peripherals essential for efficient robot control.

The fourth chapter dealt with the specific task to be implemented, the pole housing inspection station. I outlined the steps involved in designing and implementing the robot cell, detailing the specific requirements of the task and the experiences gained during the implementation. The chapter also covered the operation steps of the inspection station and the results of the testing.

The fifth chapter covered the process of robot programming. I provided a detailed explanation of the programming steps, including the application of different interpolation methods (linear and joint) for precise control of the robot's movements. I introduced the software used for programming, their functions, and the challenges encountered during programming along with their solutions.

In conclusion, the solutions and development steps presented in the thesis contribute to enhancing the efficiency of industrial automation and optimizing manufacturing processes. The experiences gained during testing and the successful handling of challenges allow for further development and application of robot systems in the future.

8. Irodalomjegyzék

- [1] IFR releases “World Robotics R&D Programs” by Carsten Heer <https://ifr.org/ifr-press-releases/news/robotics-research-how-asia-europe-and-america-invest>
- [2] China overtakes USA in robot density by Carsten Heer <https://ifr.org/ifr-press-releases/news/china-overtakes-usa-in-robot-density>
- [3] India’s Robot Boom Hits All-Time High by Carsten Heer <https://ifr.org/ifr-press-releases/news/indias-robot-boom-hits-all-time-high>
- [4] Top 5 Robot Trends 2021 by Carsten Heer <https://ifr.org/ifr-press-releases/news/top-5-robot-trends-2021>
- [5] Top 5 Robot Trends 2022 by Carsten Heer <https://ifr.org/ifr-press-releases/news/top-5-robot-trends-2022>
- [6] Top 5 Robot Trends 2023 by Carsten Heer <https://ifr.org/ifr-press-releases/news/top-5-robot-trends-2023>
- [7] Újabb mérföldkő a FANUC-nál
<https://www.cnc.hu/wp-content/uploads/2019/08/Industrial-Robots-Teaser.jpg>
- [8] Robotválasztás az intelligens gyárakhoz by Peter Lange, Business Development Manager Fixed Robotics, OMRON Europe <https://www.muszaki-magazin.hu/2021/06/11/robotvalasztas-az-intelligens-gyarakhoz/>
- [9] Comparative analysis of collaborative robots International Journal of Engineering and Management Sciences (IJEMS) Vol. 7. (2022). No. 4
<https://ojs.lib.unideb.hu/IJEMS/article/view/11985/11121>
- [10] UR5 Thecnikal Specification
https://www.usna.edu/Users/weapcon/kutzer/_files/documents/User%20Manual,%20UR5.pdf
- [11] Mathematical Modelling and Simulation of Human-Robot Collaboration
https://www.researchgate.net/publication/347021253_Mathematical_Modelling_and_Simulation_of_Human-Robot_Collaboration
- [12] Newton-Raphson iterációs eljárás egy és többszabadságfokú rendszerekre
<https://axisvm.hu/tudasbazis/szamitas-es-eredmenyek/newton-raphson-iteracios-eljaras-egy-es-tobbszabadsagfoku-rendszerekre/>
- [13] Mitsubishi RV-2FR-D-S25 Specifications
<https://hu.mitsubishielectric.com/fa/products/rbt/robot/vertical/rvfr-series/rv-2fr-d-s25.html#downloads>
- [14] Exploratory Workshop on the Social Impacts of Robotics: Summary and Issues
<https://www.princeton.edu/~ota/disk3/1982/8209/820904.PDF>

9. Ábrajegyzék

1. ábra Universal robot UR5 [10]	13
2. ábra Általam átalakított robot tesztelő/fejlesztő állomás	15
3. ábra UR5 robot paraméterek a Denavit-Hartenberg módszer szerint [11]	16
4. ábra Newton-Raphson iterációs eljárás [12]	17
5. ábra Mistsubishi Labor Tréning Robot	19
6. ábra RV-2FR befoglaló méretei [13]	20
7. ábra Mitsubishi RV-2FR-R Mozgása [13]	21
8. ábra Az ellenőrzés folyamata	23
9. ábra RT Toolbox 3 Project Szerkesztő	26
10. ábra Szimulált és Valós Robot álmomás	27
11. ábra Hand Modellek	28
12. ábra Szimulált állomások modellje	28
13. ábra Layout Tree	29
14. ábra Robot Program Waypointok	30
15. ábra Robot mozgások áramfelvétele	31
16. ábra Motion Limit Parameters	32
17. ábra Hand Parameters	33
18. ábra Súly- és méretparaméterek	34
19. ábra Ütközés elkerülés Paraméter	35

10. Táblázatjegyzék

1. táblázat Robotparaméterek táblázatban [9].....	11
2. táblázat UR5 Specifikációk [10].....	14
3. táblázat UR5 Mozgási paraméterek [10]	14
4. táblázat UR5 I/O Portok [10].....	14
5. táblázat Mitsubishi RV-2FR Specifikáció [13]	19

11. Mellékletek

1. Melléklet: Robot Program RT Toolbox 3 Project (Pick&Place Labor)