

DIGITÁLIS IKER LÉTREHOZÁSA EGY UNIVERZÁLIS CSOMAGOLÓGÉPHEZ CREATING A DIGITAL TWIN FOR A UNIVERSAL PACKAGING MACHINE

Péter Ádám ^{1*}

¹ Informatika Tanszék, GAMF Műszaki és Informatikai Kar, Neumann János Egyetem, Magyarország,

<https://orcid.org/0009-0008-3281-2392>

<https://doi.org/10.47833/2025.2.ENG.008>

Kulcsszavak:

Digitális iker
Virtuális beüzemelés
Unity
Siemens
Ipari automatizálás

Keywords:

Digital twin
Virtual commissioning
Unity
Siemens
Industrial automation

Cikktörténet:

Beérkezett 2025.július 7.
Átdolgozva 2025. szeptember 2.
Elfogadva 2025. szeptember 9.

Összefoglalás

A digitális iker technológia az ipar 4.0 egyik kulcseleme, amely lehetőséget teremt a fizikai rendszerek valós idejű szimulációjára, tesztelésére és optimalizálására. Egyetemi kutatási és fejlesztési projekt keretében egy univerzális csomagológép teljes értékű digitális ikerpárját hoztam létre, amely a Unity játékmotoron alapuló 3D környezetet, Siemens S7 PLC-kommunikációt és vizuális programozási rendszert (Playmaker) is integrál.

Abstract

Digital twin technology is a key component of Industry 4.0, enabling real-time simulation, testing and optimisation of physical systems. As part of a university research and development project, I have created a full-fledged digital twin of a universal packaging machine, which integrates a 3D environment based on the Unity game engine, Siemens S7 PLC communication and a visual programming system (Playmaker).

1. Bevezetés

A modern ipari automatizálás és digitalizáció egyik legfontosabb iránya a digitális ikrek alkalmazása, amely lehetőséget biztosítanak a gépek és berendezések valós idejű szimulációjára, optimalizálására és távoli felügyeletére. Egy egyetemi projekt, illetve pályázat keretében célom volt egy univerzális csomagológép digitális ikerének kifejlesztése, amely nemcsak a fizikai berendezés pontos digitális mását biztosítja, hanem lehetőséget teremt annak működésének elemzésére, tesztelésére és fejlesztésére is. A megoldás révén a gyártási folyamatok hatékonysága növelhető, a karbantartási költségek csökkenthetők és a termelési hibák minimalizálhatók. A digitális iker technológia alkalmazása napjainkban már nem csupán a nagyvállalati gyártórendszerek privilégiuma, hanem egyre inkább elérhető az oktatás, a kutatás-fejlesztés és a kis- és középvállalkozások számára is. A valós idejű adatszinkronizálás, a 3D szimuláció és a virtuális beüzemelés révén nemcsak a gyártás hatékonysága növelhető, hanem a hibakeresés, a folyamatfejlesztés és az oktatás is biztonságosabbá és költséghatékonyabbá válik.

A cikk hátralevő részében a második szakasz a digitális iker technológia elméleti alapjait és ipari hátterét ismerteti, míg a harmadik szakasz a Unity játékmotor szerepét és jellemzőit mutatja be a rendszer felépítésében. A negyedik szakasz részletesen bemutatja az univerzális csomagológép felépítését, az ötödik szakasz pedig a digitális iker gyakorlati megvalósítását és a virtuális beüzemelés folyamatát ismerteti. [1]

* Kapcsolattartó szerző, email: peter.adam@nje.hu

2. Digitális iker bemutatása

Az első digitális iker a NASA-nál jött létre az 1960-as években, az Apollo-missziók modellezésének eszközeként használták fel. A NASA szimulátorként alkalmazta az Apollo 13 oxigéntartályainak meghibásodásának tesztelésére. A digitális ikreket David Gelernter 1991-es Mirror Worlds (Tükörvilágok) című könyve vetítette előre.

A digitális iker koncepciója több részből is állhat:

- A fizikai tárgy vagy folyamat és annak fizikai környezetének digitális reprezentációja
- Fizikai és a virtuális reprezentációk közötti kommunikációs csatorna

A digitális iker valójában egy új generációs komplex rendszerszintű tervezési módszertan, amivel nem csupán a 3D-s virtuális grafikai objektumok modellezése tartozik, hanem a folyamatok tervezése és nyomon követése, a különböző CPU-val ellátott vezérlők programozása. Ide tartozik a PLC programozás, az ipari robotok vezérlése is. Tehát a digitális iker meglehetősen multidiszciplináris tulajdonságokkal rendelkezik.

Ha a digitális ikert a hagyományos célgép és robotcella, gyártósor tervezés oldaláról kezdjük el megközelíteni, akkor a következő szintek szerint lehet megvalósítani. [5]

A digitális iker pontossági szintjei:

- Statikus modell
- Kinematikai modell
- Kinetikai modell
- Procedurális fizikai, animációs és tönkremeneteli modell
- Fotorealisztikus modell

Statisztikus modell szint: a hagyományos 2D/3D terméktervező szoftverek által létrehozott objektum(ok), amelyek csak hálóval és anyaggal vannak ellátva, esetleg még texturával. A geometriai, koordináta rendszer adatokon kívül egyéb attribútumokat is tartalmazhat, melyek nem feltétlenül szükségszerűek a digitális iker szempontjából.

Kinematikai modell szint: Ezen a szinten a modell mozgásviselkedésének alapvető leírása történik meg, elsősorban translációs és rotációs mozgásokat végző aktuátorok mozgásának animációjára összpontosítva. A mozgás paraméterei – mint például a sebesség, gyorsulás/lassulás, kezdő- és végpozíciók, valamint mozgáspályák – manuálisan vagy procedurálisan is módosíthatók.

Fontos megjegyezni, hogy a kinematikai modell is tartalmaz megkötéseket, melyek az egyes elemek relatív szabadságfokait szabályozzák (pl. csuklók, korlátok). Ezek a megkötések lehetővé teszik a rendszer fizikai koherenciájának fenntartását, még akkor is, ha a fizikai motor ebben a szakaszban inaktív.

A kinematikai modellezés során gyakran alkalmazott struktúra például a URDF (Unified Robot Description Format), amely a modell hierarchiáját és mozgásláncát írja le, és könnyen integrálható különböző szimulációs környezetekbe, mint például Unity vagy ROS-alapú rendszerek.

Kinetikai modell szint: A kinetikai szinten már a fizikai motor is aktív szerepet kap, így a modell elemei valódi fizikai kölcsönhatásokba léphetnek egymással, szemben a pusztán előre definiált mozgásokkal. Itt lehetőség nyílik erők, nyomatékok, tömegek, súrlódási tényezők és fizikai kényszerek definiálására, amelyek segítségével valósághű fizikai szimuláció valósítható meg.

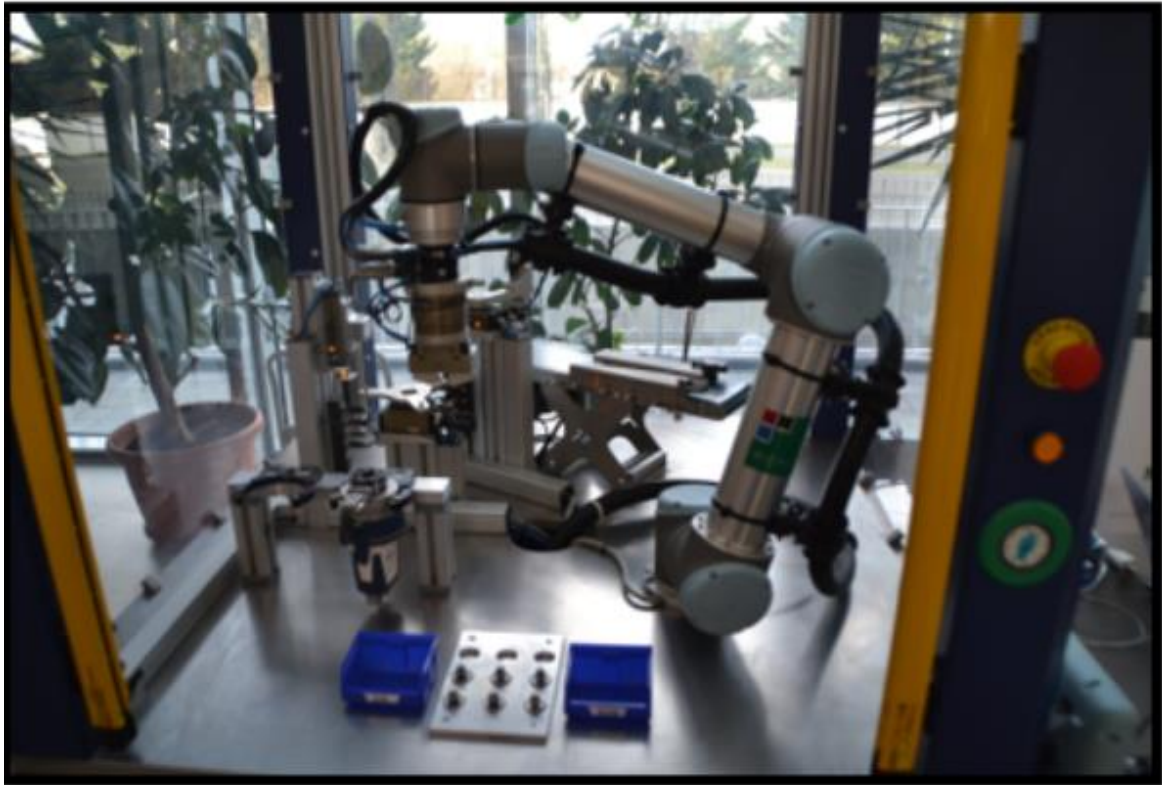
A Unity beépített fizikai motorja (például PhysX) lehetővé teszi ezeknek a kölcsönhatásoknak a valós idejű szimulációját, ami a digitális ikerpár hiteles viselkedéséhez elengedhetetlen. Ebben a szakaszban a korábban definiált kinematikai szerkezetek és megkötések továbbra is érvényben maradnak, azonban a rendszer viselkedése már dinamikusan, a fizikai szabályok szerint alakul.

Procedurális modell szint: ezen a szinten tönkremeneteli modelleket is tudunk kezelni. Mint például az ütközések hatására bekövetkező deformációk szimulálása. Valamint még élethűbb fizikai kölcsönhatások létrehozása. Illetve főként az organikus jellegű objektumoknak a dinamikus mozgás

animációk lekezelése. Például: organikus és nem organikus humanoid operátorok főbb mozgásainak és mozdulatainak animációs vezérlése.

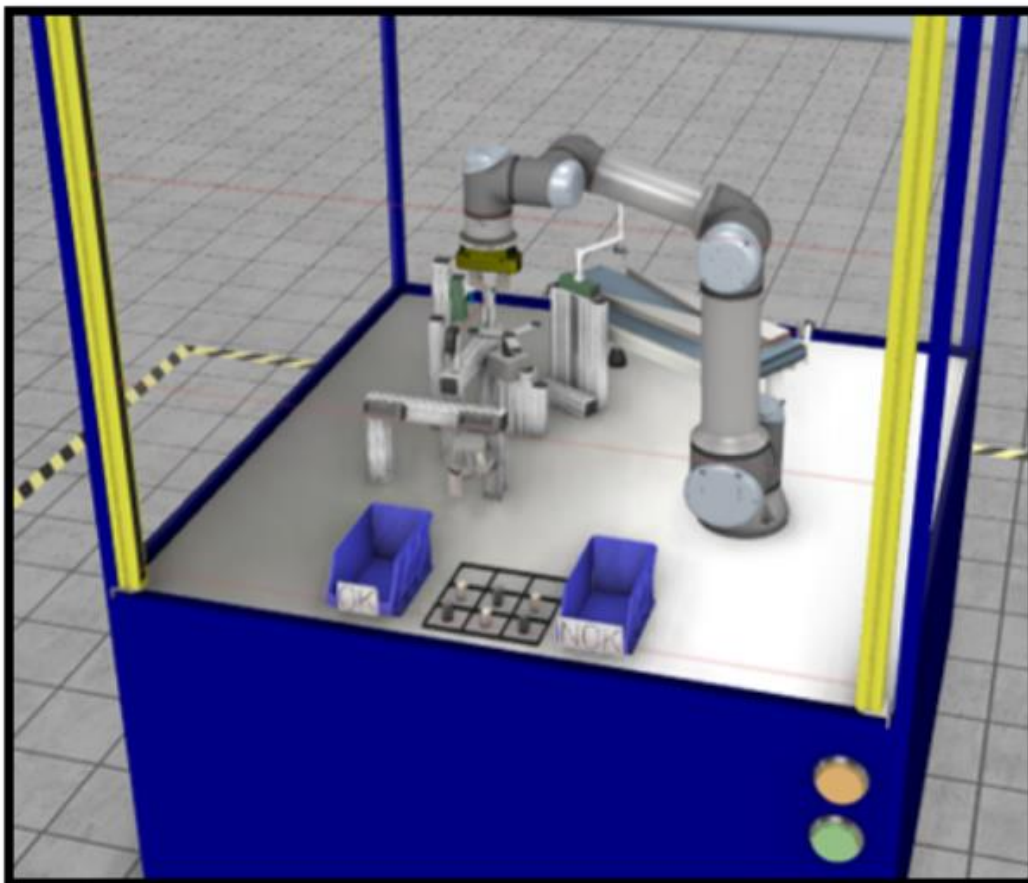
Fotorealistikus modell szint: előző szinteket kiegészítjük a “fotoreal” 3D grafikai megjelenítéssel, ezáltal élethű virtuális környezetet tudunk létrehozni.

Alapvetően a digitális iker készül el egy adott célgépnek vagy gyártósornak először és majd ezután fog következni a gyártási fázis, ahol a valódi célgép fog létrejönni. Viszont ez a folyamat történhet vissza felé is azaz “Reverse engineering” módszer segítségével rekonstruáljuk a már létező gép 3D modelljeit. Ezt követően pedig már meg lehet valósítani a teljes virtuális rekonstrukciót. Az első ábrán látható egy valós célgép, a második ábrán az utólagos digitális ikerpárja látható.



1. Ábra. 2022 PLC verseny második verseny feladatának a célgépe

A 2. ábrán látható a célgép digitális ikertestvére, amely a Visual Components szoftverrel készítettem el.



2. Ábra. Az 1. ábrán látható célgép digitális ikerpárjának egyszerűsített fényképek alapján rekonstruált változata

3. Unity játékmotor bemutatása

Digitális ikerpárok készítésére több szoftver is alkalmas lehet, a Unity játékmotor lett kiválasztva az univerzális csomagológéphez.

A Unity egy korszerű, rugalmas játékfejlesztő motor, amely lehetővé teszi a fejlesztők számára, hogy különböző platformokra, mint a PC, konzolok, mobil eszközök és VR rendszerek, professzionális játékokat és alkalmazásokat hozzanak létre. Rugalmas eszközkészlete és fejlett grafikai motorja miatt a kezdőktől a profi fejlesztőkig sokan használják.

A Unity Technologies 2005-ben adta ki először azt a játékmotort. Kezdetben csak macOS-re fejlesztették, de hamar kinőtte magát egy teljes körű, könnyen elérhető játékfejlesztő platformmá, amely ma már több száz platformot támogat. Az évek során a Unity folyamatosan fejlődött és ma már a játékipar egyik legfontosabb motorjává vált.

A Unity főbb jellemzői:

- **Multiplatform támogatás:** A Unity lehetővé teszi a játékok egyszeri fejlesztését és több platformra történő exportálását, beleértve a Windows, macOS, Linux, iOS, Android, PlayStation, Xbox, Nintendo Switch és VR/AR eszközöket.
- **Grafikai motor:** A Unity fejlett renderelési technológiával rendelkezik, amely lehetővé teszi a fotorealistikus és stilizált grafikák megvalósítását egyaránt. Támogatja a HDRP (High Definition Render Pipeline) és a URP (Universal Render Pipeline) technológiákat.
- **Programozás és szkriptelés:** A Unity a C# programnyelvet használja, amely egy hatékony és rugalmas nyelv. Az integrált fejlesztőkörnyezet (IDE) közé tartozik a Visual Studio és a Rider is.

- Fizikai motor: A Unity beépített fizikai motorja (NVIDIA PhysX) segítségével pontos és valósághű fizikai szimulációkat lehet készíteni.
- Animációs rendszerek: Az Animator és Timeline rendszerek lehetővé teszik karakterek és környezeti objektumok animálását és dinamikus jelenetek létrehozását.
- Közösségi támogatás: A Unity hatalmas felhasználói bázissal rendelkezik, amely segíti az új fejlesztőket tanfolyamokkal, fórumokkal és dokumentációval.

A 3. ábrán látható egy célgép, amit Unity-vel készítettem el, ami a 2024. Országos Ajtonyi István Irányítástechnikai Programozó Versenynek az egyik PLC gyakorlati versenyfeladatnak a rekonstrukciója. A modellt a versenyre való felkészülés céljából lett elkészítve. [4]



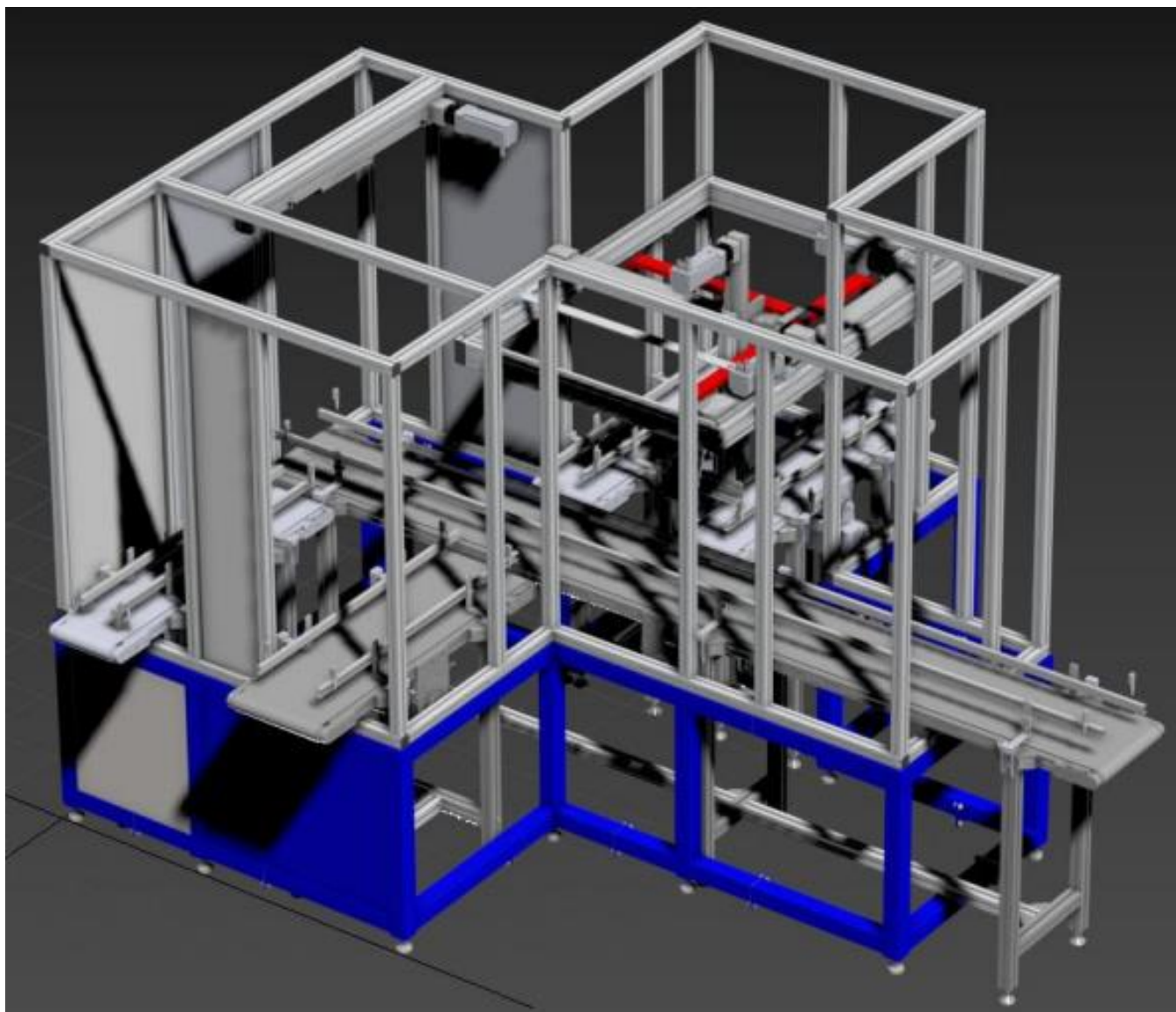
3. Ábra Unity-vel készült PLC verseny feladat rekonstrukciója

4. Univerzális csomagológép bemutatása

A csomagológép a 2020-1.1.2-PIACI-KFT-2020-00062 számú pályázat keretén belül jött létre és ennek a 3D modelljéhez kell elkészíteni a digitális ikret, amit STEP formátumban küldtek meg, ami egy általános 3D modell fájl formátum.

A célgép fő feladata különböző méretű termékek és dobozok palettázása, pakolása.

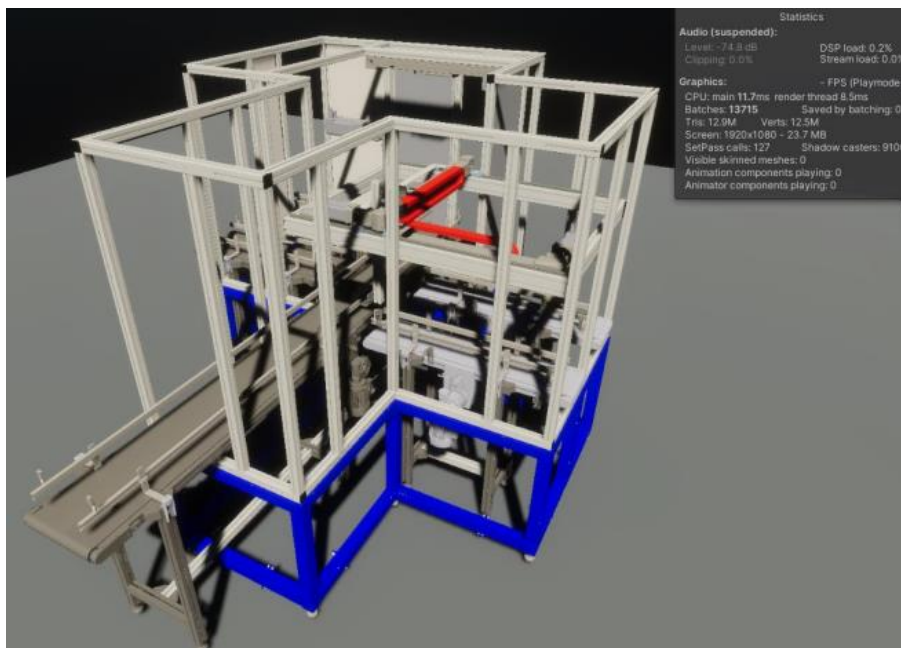
A gép főbb elemei 5 darab futószalag pálya, amiken a bemeneti és kimeneti anyagáramlás történik. A következő fontosabb komponense egy 3D mozgásokra képes X-Y-Z translációs csuklókkal rendelkező manipulátor. Decartes-féle koordináta-rendszerrel rendelkezik. Továbbá a mozgatott robotszerszám egy vákuum megfogó. A 4. ábrán a csomagológép 3D modellje látható 3Ds Max szoftverbe importálva.



4. Ábra a csomagológép 3d modellje

5. Digitális ikerpár létrehozása

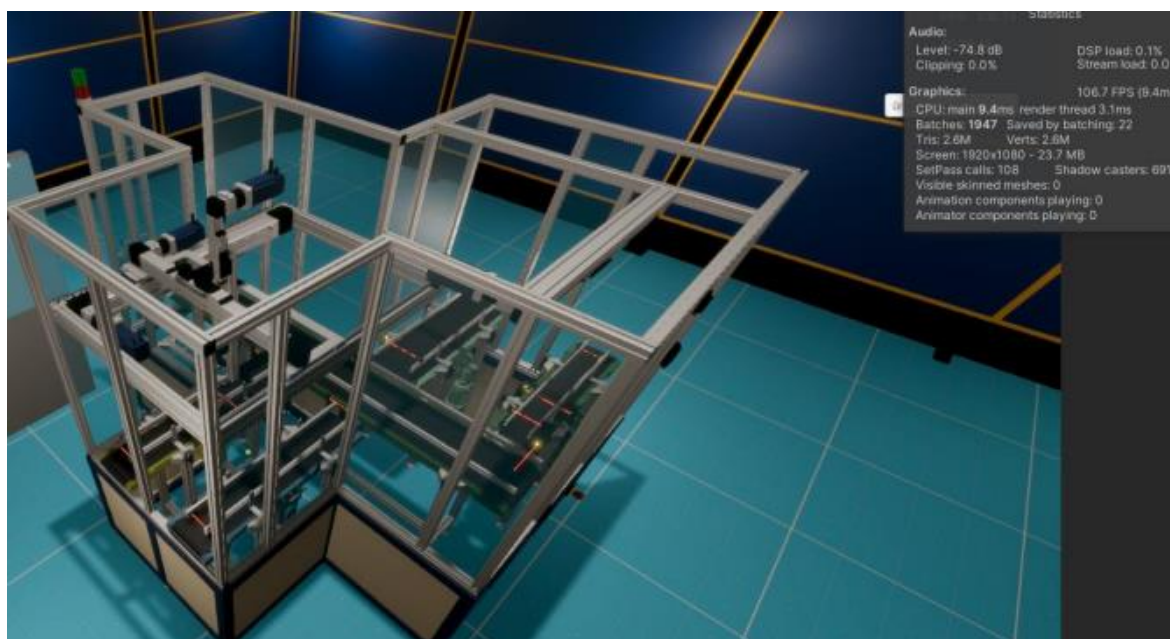
A Unity personal licensze alapvetően az FBX fájl formátumot támogatja leginkább, ezért a kapott modellt első lépésként erre formátumra kellett átkonvertálni. Az Autodesk 3Ds Max segítségével történt. Az átkonvertált modell fájl formátuma FBX, az így kapott modellt kell a Unity project könyvtárba beimportálni. Ez után a modell optimalizálását kell elvégezni. A folyamat történhet már a 3ds Max-al is vagy a Unity-n belül is elvégezhető. Az optimalizációs folyamat során elsősorban a modell poligonszámának a vizsgálatát kell elvégezni, mert ez jelentősen befolyásolni tudja a szimuláció futási teljesítményét. Alapértelmezetten az általános termékfejlesztői rendszerekkel készült modellek topológiája meglehetősen részletes és ezt automatikusan hozza létre az adott rendszer. Ezzel nem igazán foglalkoznak egy célgép vagy az adott gépelem tervezése és modellezése során a konstruktőrök. A Unity háromszög alakú poligonokból jeleníti meg az objektumok hálóját. A 5. ábrán látható már a Unity-be beimportált csomagológép 3D modellje, illetve a főkamera által megjelenített képe, illetve a statisztikai debug ablakban látható a "Tris:" (poligonszám) is.



5. Ábra a Unity-be importált modell az optimalizálás előtt 12.9 millió darab poligon

A kép alapján egyértelműen látszik, hogy rendkívül sűrű hálóval rendelkezik a csomagológép. Körülbelül 12,9 millió poligonból épül fel, ami a célgép alkatrészeinek bonyolultságához képest meglehetősen sok. Ezt mindenképpen csökkenteni kell. Azokat az objektumokat, amik a szimuláció futása során a virtuális célgépnek a működését nem befolyásolják azok törölhetők. A futási teljesítményt jelentősen ronthatják azok a modellek, amelyek működés közben nem láthatók. Ilyen gépelemek például az alátétek, csavarok, egyéb kötőelemek és belső szerkezeti elemek. Természetesen, ha olyan igény merül fel, amely megköveteli ezek jelenlétét — például összeszerelési vagy szétszerelési szimuláció esetén —, akkor minden alkatrészt megtartunk.

A következő 6. ábrán látható a már optimalizált csomagológép 3D modelljei.



6. Ábra Unity-ben már optimalizált csomagológép modellje kb. 2.6 millió darab poligon

A jelenet kamerája által megjelenített poligonszám a 6. ábrán körülbelül 2.6 millió darab, ebbe beletartozik a környezet vizualizációja is.

A modell optimalizálása után a következő lépés az objektumok hierarchiájának beállítása. A Unity-in belül az adott jelenethez tartozik alapértelmezetten a szoftver kezelői felületének bal oldalán található a "Hierarchy" fül. Amely tartalmazza a jelenet összes objektumát. A jelenet neve "Pályázat_csomagológép" négy darab fő szülő játékobjektumot tartalmaz:

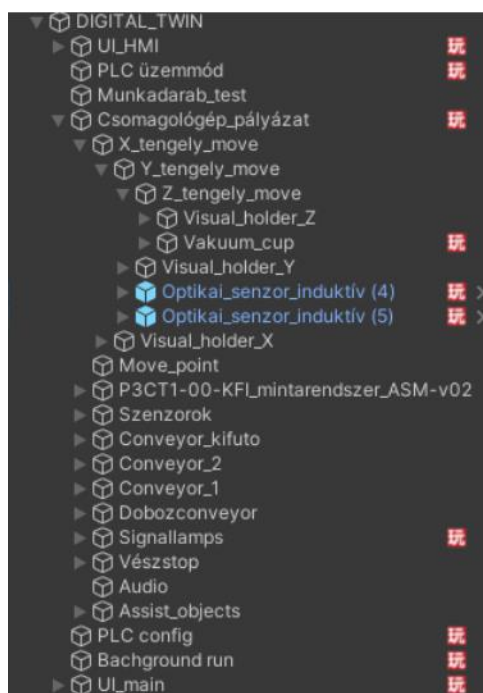
- LEVELMASTER
- STATIKUS_ELEMEK
- JÁTÉKOS_TÁROLÓ
- DIGITAL_TWIN

A "LEVELMASTER" objektum tartalmazza a jelenetet támogató komponenseket, mint például a fényeket, a vizuális megjelenítést tartalmazó komponenseket, egyéb menedzser játékobjektumokat a Siemens S7-es PLC-kel kommunikációért felelős objektumokat és komponenseit.

A "STATIKUS_ELEMEK" a jelenet statikus díszítő és környezeti elemeit, modelljeit tartalmazza elsősorban. Továbbá itt vannak létrehozva a fizikai ütköző dobozok is. Amelyek a játékost a játéktéren belül fogják tartani.

A "JÁTÉKOS_TÁROLÓ" egy fps (first-person) kamera nézetű játékos vezérlő objektumait és a fő kamerát tartalmazza. A játékos segítségével lehet interakcióba kerülni a digitális ikerrel. Ez főként a 3D HMI-in keresztül fog megtörténni.

A "DIGITAL_TWIN" tartalmazza magát a célgépet és hozzá tartozó segédobjektumokat. A DIGITAL_TWIN objektum részletes bemutatása. A 7. ábrán az objektum struktúrája látható.



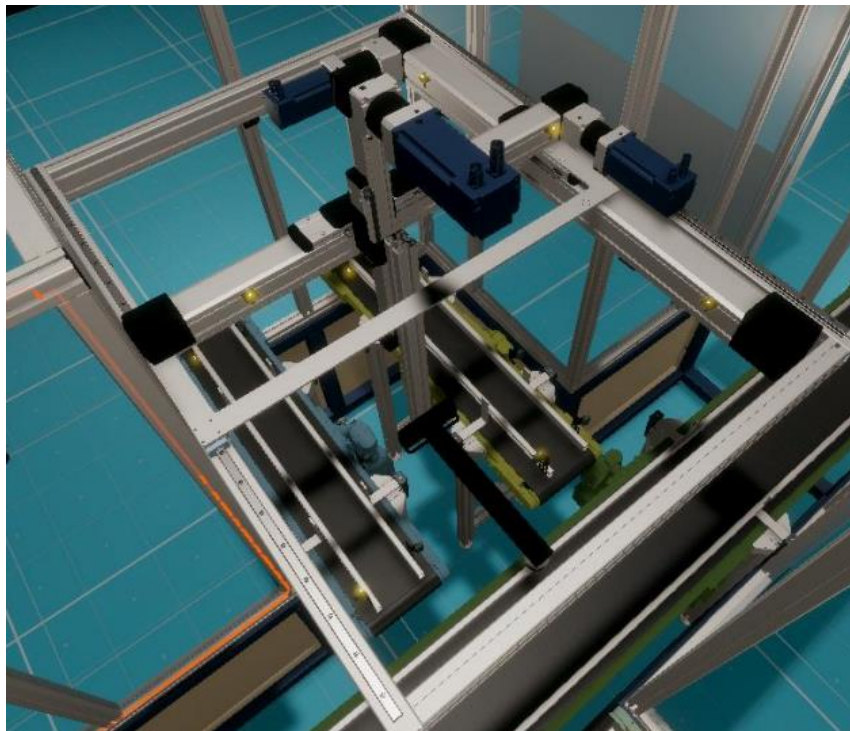
7. Ábra "DIGITAL_TWIN" játékobjektum felépítése

Az első gyerek objektum a UI_HMI, amely a felhasználói vezérlő felületeket tartalmazza. A főbb funkciók:

- Start-Exit
- 3D hajtásrendszer Jog üzemmód (kézi üzemmód)
- Futószalagok optikai szenzor felügyelet
- Hajtásrendszer végállás szenzor felügyelet
- Futószalagok motor felügyelet
- Csomag, rendelés összeállító felület
- Szenzorok mérés beállító felület

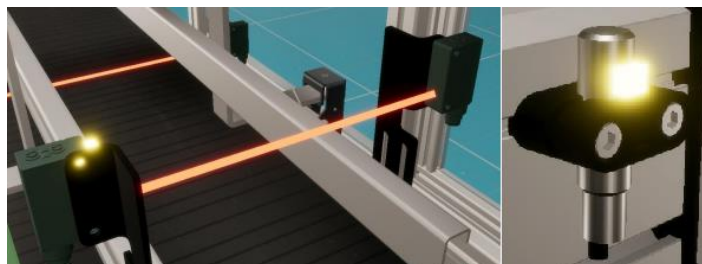
A "Csomagológép_pályázat" objektum bemutatása (7. Ábra), ez tartalmazza a célgép grafikai, működtető segéd objektumait és komponenseit, illetve a vezérlő PlayMaker FSM (Finite-State Machine) kódjait.

A célgép egyik legkomplexebb komponense a 3D hajtásrendszer és a rajta lévő vákuum megfogó. Ez fogja végrehajtani a termékek és a dobozok mozgatását és pozicionálását. A részletes felépítése a következő ábrán látható majd. Itt lényeges a megfelelő úgynevezett szülő-gyerek kapcsolatot létrehozni, hogy a mozgás animáció jelleghelyes legyen. A működéshez hozzá kapcsolódik továbbá egy "Move_point"-nak elnevezett üres gameobject is, amely tekinthető egyfajta programozott pontnak is. Ha valós időben módosítjuk a Move_point transzformációs mátrixának első sorát az az a X; Y; Z koordinátáinak pozíció értékeit megváltoztatjuk adott intervallumon belül, akkor a három tengely azonnal reagálni fog és fel fogja venni a Move_point pozíciójának értékeit. Ezzel létre tudjuk hozni a hajtásrendszer mozgás animációját. A hajtásrendszer Unity-ban felépített képét a következő 8. ábra mutatja meg.



8. Ábra hajtás rendszer képe a Unity-ban

A következő fontosabb elemek az optikai helyzetérzékelők és az induktív végállás érzékelők. Ezek működés szempontjából hasonlóan épülnek fel. A Unity-s felépítés alapján a különbség csupán annyi, hogy az optikai szenzoroknál plusz egy grafikai effekt vezérlő kód is szerepel, mely egy fénynyaláb megjelenítésért felel. A következő 9. ábra mutatja meg a felhasznált szenzorok grafikai kinézetét a Unity belül.



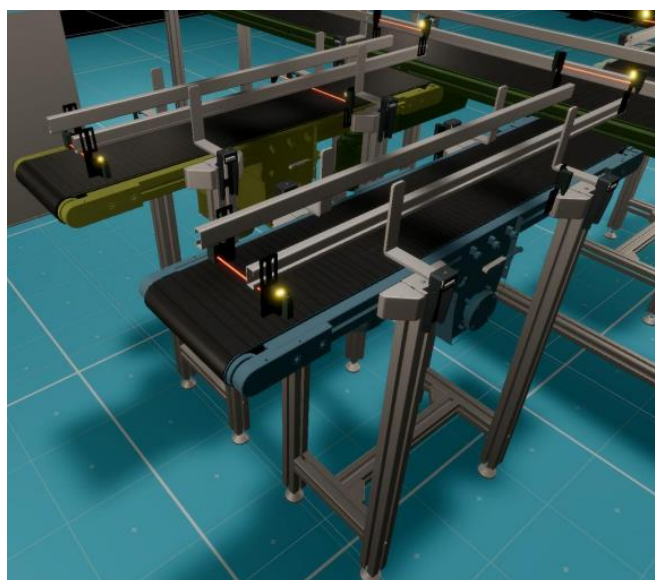
9. Ábra futószalagokon lévő optikai szenzor és a hajtás rendszer véghelyzet szenzor képe

Egy virtuális futószalag megvalósítása relatív komplex feladatnak minősül. A hierarchiai felépítése [10.ábra], viszont meglehetősen egyszerű akár egyetlen egy objektummal is megvalósítható. A célgépnél meglévő pálya felépítését a következő 10. ábra mutatja be. Alapvetően egy fő objektumból áll és az első gyerek objektuma "NONE" néven tartalmazza a teljes grafikai modell hálót. Ez nem feltétlenül szerencsés, de a forrás modellnél is így épült fel. Továbbá négy darab gyerek objektummal kiegészítésre került. Ezek a futószalag audiovizuális hatása miatt kellenek. A vezérlő kódok is a "NONE" objektumon találhatóak.



10. Ábra egy futószalag felépítése

A "NONE" objektumon az FSM kódok mellett van három további lényeges komponens, amik a munkadarabok vagy termékek fizikai szállításáért felelős. Ez kettő darab Box Collider amiből az egy trigger zónaként működik. A másik Box Collider pedig a fizikai ütközőként funkcionál. Továbbá egy Rigidbody komponenst is tartalmaz, aminél kinematikai funkció van bekapcsolva. Az első vezérlő kód a "Munkadarab kezelés" nevet viselő FSM kód. Ez felelős a pályára érkező munkadarabok eltárolására, elindítására, illetve a kilépő termékek kezelését is végzi. A következő 11. ábra a Unity-ben felépített konveor pályákat mutatja meg.



11. Ábra a konveor pályák képe

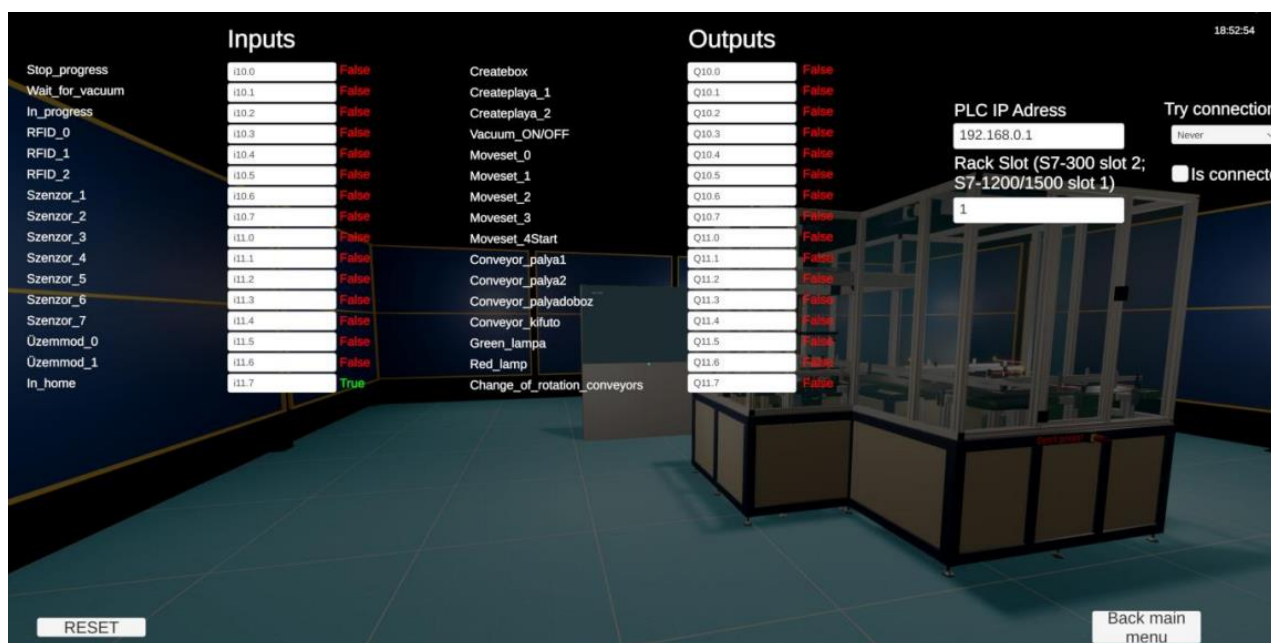
5. Virtuális beüzemelés

Az univerzális csomagológép a Siemens SIMATIC rendszerekkel kompatibilis a PLC programot a v17 TIA portallal készült el egy S7-1214C típusú PLC-re. A következőkben a PC-PLC kapcsolat felépítése lesz bemutatva. Ehhez a realvirtual.io S7Interface prefab-ot használtam fel (ami a Unity belül egyfajta sablon), illetve hozzá tartozó komponenseit. A realvirtual.io egy fejlett digitális iker és virtuális üzembe helyezési keretrendszer, amely a Unity játékmotorra épül. Célja, hogy valósághű 3D szimulációkat és vizualizációkat biztosítson a gépészeti és automatizálási ipar számára. A rendszer lehetővé teszi a gépek és rendszerek valós idejű működésének modellezését, szimulációját és tesztelését, mielőtt azok a fizikai világban megvalósulnának.

A LEVELMASTER fő objektumnak első gyerek objektuma a realvirtual, ami tartalmaz egy Realvirtual Controller nevezetű scriptet, ez kell a PC-PLC kapcsolat működéséhez. A realvirtual gameobjecthez hozzá adtam két prefabot ami a S7interface. Ennek a prefabnak egy komponense van, az S7 Interface script, aminek lehet megadni, hogy milyen állapotban legyen a kommunikációs

csatorna, illetve IP címet is itt lehet megadni. Továbbá IO összerendelési táblázatokat is lehet beimportálni. A rendszer és a PC között S7-1200 PLC teremt kapcsolat. Kettő darab S7Interface objektum van, az "1"-re végződő nevű objektum kezeli a PLC bemeneti változóit. Ezt a Unity fogja felülrni és a PLC pedig olvasni fogja. A "2"-re névre végződő pedig a PLC kimeneti változóit fogja kezelni. Ez annyit jelent, hogy a Unity fogja ezeket olvasni a PLC pedig felülrni.

Ahhoz, hogy könnyedén tudjuk kezelni a PC-PLC kapcsolatot érdemes készíteni egy UI felületet, ahol a bemeneti és kimeneti változók címeit meg lehet adni. (A PLC típusát, illetve IP címét) Ezt a felhasználói felületet az Esc gomb lenyomásával lehet előhívni, újbóli megnyomásával pedig be lehet zárni. A következő ábra a felhasználói felületet mutatja be. Ahol a nevezetes változók nevei és állapotuk látható, illetve a PLC címük is. A PLC IP Adress-nél kell megadni a PLC IP címét. A Try connection-nél pedig lehet állítani egy legördülő menü segítségével, hogy mindig csatlakozzon a PLC-re vagy ne. A Rack Slot ha S7-1200/1500 PLC van akkor 1-est kell beírni, ha pedig S7-300/400, akkor 2-est kell beírni. A RESET gombbal pedig újra tudjuk indítani a jelenetet. A Back main menü megnyomásával pedig a fő menübe tudunk visszalépni. A 12. ábra ezt a UI felületet mutatja be.



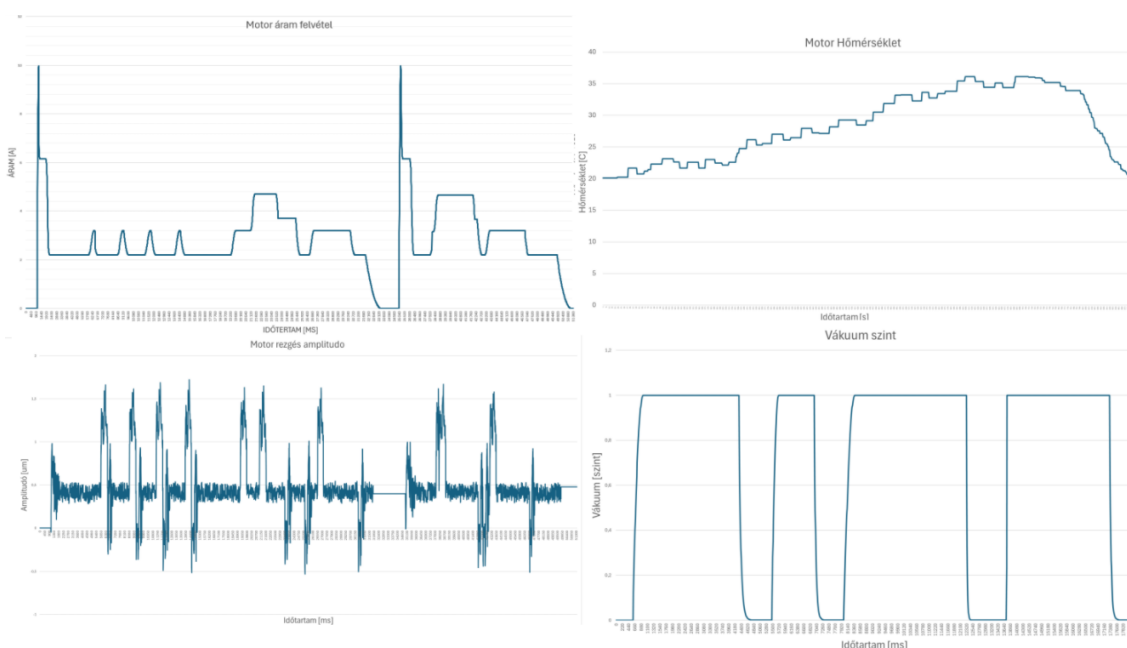
12. Ábra a PLC konfiguráló felhasználói felület

6. Mérések

A már korábban bemutatott felhasználói felületeknél van lehetőségünk méréseket végezni. A működés lényege, hogy egy adott mintavételezési frekvenciával egy kívánt változó értékét gyűjti egy listába. Ameddig tart a mérés ezt ciklusonként végrehajtódig, mentéskor pedig a megadott helyre egy excel.csv fájl fog létre jönni.

A digitális iker igazából szimulált értékekkel dolgozik. Ami lehet egy előre definiálható determinisztikus függvény is, illetve teljesen véletlenszerűen generált. Lehetőség van a kettő ötvözetének a létrehozására is. Viszont a PC-PLC kapcsolat segítségével a PLC-ből van lehetőség, hogy valós szenzor adatokat tudunk kiolvasni és azokat kimenteni. A virtuális valóságnak köszönhetően bármilyen matematikailag leírható fizikai mennyiség virtuális mérését lehet szimulálni

Négyféle mérést készítettem el a futószalagpályák villanymotorjainak az áramfelvételét, forgórészének rezgését és a motor hőmérsékletét lehet mérni, valamint a vákuum megfogó vákuum szintjének a meglétét lehet még elmenteni. Ezek mind szimulált mérések, amiket a Unity hajt végre. Ezeket a méréseket nem kell figyelembe venni, mert nem is az volt a cél, hogy a valóságnak megfelelő eredmények keletkezzenek, hanem az adatgyűjtő és csv. fájl mentéséért felelős programkódok tesztelésért jöttek létre. A következő 13. ábrán egy-egy mérés képe látható.



13. Ábra virtuális mérés képei balfelső motor áramfelvétel, jobbra fent motor hőmérséklet balra lent a motor rezgés amplitúdója, jobbra lent a vákuum szint változás látható

7. Következtetések

A project célkitűzése, miszerint egy univerzális csomagológép digitális ikerpárjának létrehozása történjen meg egy interaktív 3D környezetben, sikeresen megvalósult. A projekt során a Unity játékmotor, a Siemens S7 PLC és a Playmaker vizuális programozási eszköz integrációja révén egy olyan komplex digitális iker jött létre, amely képes valós idejű szimulációra, tesztelésre és vizuális visszacsatolásra.

A fejlesztés jelentősége több szinten is értelmezhető. Egyrészt a kutatás egy kézzelfogható példát szolgáltat arra, hogyan lehet a játékipar technológiáit – különösen a Unity-t – hatékonyan alkalmazni az ipari automatizálás és a virtuális beüzemelés területén. Másrészt a rendszer segítségével lehetővé válik a célgépek működésének mélyebb megértése, a hibák előzetes feltárása, valamint a karbantartási és fejlesztési költségek csökkentése.

Tudományos szempontból a projekt hozzájárul az ipar 4.0 keretében zajló digitális transzformációhoz, mivel demonstrálja a digitális iker technológia gyakorlati alkalmazását, és ezzel elősegíti az elmélet és a gyakorlat közötti szakadék áthidalását. A munka multidiszciplináris jellege – amely magában foglalja a gépészeti modellezést, automatizálást, 3D vizualizációt és szoftverfejlesztést – további lehetőségeket nyit más gépipari vagy oktatási célú alkalmazások számára is.

Összességében a kutatás bizonyítja, hogy egy digitális iker nemcsak egy egyszerű vizuális modell, hanem egy funkcionálisan integrált, valós idejű szimulációs rendszer is lehet, amely hatékony eszköz lehet a tervezés, oktatás, szervizelés és gyártás támogatásában.

Köszönetnyilvánítás

A szerzők köszönetet mondanak a projektben résztvevő intézmények - Nádor Rendszerház Kft., Controlsoft Automatika Szolgáltató Kft., Neumann János Egyetem GAMF Műszaki és Informatikai Kar - kollégáinak. Köszönettel tartozunk a kutatás támogatásáért, amely az "Ipar 4.0 moduláris felépítésű ipari csomagológép fejlesztése integrált adatelemzéssel és mesterséges intelligenciára épülő optimalizálással, hibaelemzéssel 2020-1.1.2-PIACI-KFI-2020-00062" pályázat keretében valósult meg. A projekt a Magyar Állam és az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával, a Széchenyi 2020 program keretében valósul meg.

Irodalomjegyzék

- [1] Industry 4.0 Policy Department Economic and Scientific Policy, 2016, p 22-23
- [2] Harshama, F.;Tomizuka, M., Fukda, T., Mechatronics-What is it; why and how? 1996
- [3] TIA Help Viewer
- [4] <https://docs.unity3d.com/6000.1/Documentation/Manual/UnityManual.html> Unity Manual utolsó megtekintés 2025.07.04
- [5] https://en.wikipedia.org/wiki/Digital_twin Digital twin utolsó megtekintés 2025. 05.05
- [6] Radanliev, P. et al. (2022) Digital twins: artificial intelligence and the IoT cyber-physical systems in Industry 4.0. Int. J. Intell. Robot. Appl., 6, 171–185. DOI:10.1007/s41315-021-00180-5
- [7] Huang, Z. et al. (2021) A Survey on AI-Driven Digital Twins in Industry 4.0: Smart Manufacturing and Advanced Robotics. Sensors, 21(19), 6340. DOI:10.3390/s21196340
- [8] Javaid, M.; Haleem, A.; Suman, R. (2023) – Digital Twin applications toward Industry 4.0: A Review. Cognitive Robotics, 3, 71–92. DOI:10.1016/j.cogr.2023.04.003