

# Program

## Digitalna transformacija robotiziranih tovarnih prihodnosti

### DIGITOP

#### **Rezultat 5.3:**

Osnovni digitalni dvojček proizvodnega sistema LabTOP

|                             |                                                                |
|-----------------------------|----------------------------------------------------------------|
| <b>Projekt/RRP</b>          | DIGITOP/RRP5, Aktivnost 5.2                                    |
| <b>Vrsta dokumenta</b>      | Poročilo                                                       |
| <b>Verzija dokumenta</b>    | 1.0                                                            |
| <b>Sodelujoči partnerji</b> | /                                                              |
| <b>Avtorji</b>              | Dr. Krištof Debeljak, Dr. Janez Povh, Vinko Longar, Rok Struna |
| <b>Datum priprave</b>       | 10.7.2025                                                      |

## Kazalo

|          |                                                                                                                            |           |
|----------|----------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>1</b> | <b>UVOD .....</b>                                                                                                          | <b>3</b>  |
| <b>2</b> | <b>OPIS SISTEMA.....</b>                                                                                                   | <b>3</b>  |
| <b>3</b> | <b>OPIS IZDELAVE DIGITALNEGA DVOJČKA .....</b>                                                                             | <b>4</b>  |
| 3.1      | MODELIRANJE KOMPONENT .....                                                                                                | 4         |
| 3.2      | POVEZAVA Z REALNIM SISTEMOM PREKO OPC UA.....                                                                              | 5         |
| 3.3      | SISTEM ZA SPREMLJANJE PORABE ENERGIJE .....                                                                                | 6         |
| 3.4      | POTEK DELA/MODELIRANJE PROCESOV IN ZAGOTAVLJANJE SKLADNOSTI .....                                                          | 7         |
| 3.4.1    | <i>DOLOČANJE ŠTEVILA PONOVIČEV CIKLA TER IZBIRO ROBOTSKIH PROGRAMOV TER ZAPOREDJA PROCESOV S POMOČJO SPREMENLJIVK.....</i> | <i>8</i>  |
| <b>4</b> | <b>USPOSABLJANJE OPERATERJEV V VIRTUALNEM OKOLJU .....</b>                                                                 | <b>9</b>  |
| 4.1      | UVOZ LABORATORIJSKEGA OKOLJA V UNITY .....                                                                                 | 9         |
| 4.2      | USPOSABLJANJE OPERATERJA NA POSAMEZNI CELICI.....                                                                          | 10        |
| <b>5</b> | <b>BIBLIOGRAFIJA .....</b>                                                                                                 | <b>14</b> |

## 1 UVOD

Digitalna preobrazba industrije, kot jo opredeljujeta koncepta Industrije 4.0 in 5.0, temelji na uporabi naprednih virtualnih orodij za optimizacijo proizvodnje, nadzor procesov in izboljšanje usposobljenosti kadrov. V tem kontekstu se digitalni dvojček uveljavlja kot ključno orodje, saj predstavlja virtualno zrcalno sliko realnega sistema, ki omogoča preizkušanje različnih scenarijev, iskanje optimizacijskih možnosti in varno usposabljanje operaterjev. V okviru projekta DIGITOP smo v aktivnosti 5.2 izdelali osnovni digitalni dvojček proizvodnega sistema laboratorija LabTOP, pri čemer smo uporabili programsko orodje Visual Components Professional. Cilj aktivnosti je bil izdelati natančen in učinkovit virtualni model proizvodne linije, ki bi omogočal razvoj, testiranje in izobraževanje, obenem pa zagotovil podporo nadaljnji digitalizaciji procesov.

V sodobnem industrijskem okolju implementacija digitalnih dvojčkov predstavlja ključno konkurenčno prednost, saj omogoča zmanjšanje tveganja pri uvedbi novih tehnologij, zmanjšanje stroškov vzdrževanja in hitrejše odločanje na podlagi realnih podatkov. Poleg tega digitalni dvojčki omogočajo napredno analizo podatkov, predvidevanje napak in prilagajanje proizvodnih procesov v realnem času, kar je v skladu s težnjami industrije po večji fleksibilnosti in trajnostni učinkovitosti. V okviru LabTOP sistema digitalni dvojček poleg izboljšanja nadzora in optimizacije omogoča tudi simulacijo različnih učnih scenarijev za operaterje, kar bistveno prispeva k dvigu kompetenc zaposlenih in varnejšemu delovnemu okolju.

## 2 OPIS SISTEMA

Laboratorij LabTOP predstavlja sodoben eksperimentalni proizvodni sistem, ki omogoča simulacijo realnih industrijskih procesov v nadzorovanem in prilagodljivem okolju. Zasnovan je bil kot učna platforma za demonstracijo konceptov tovarn prihodnosti, razvoj inovacij na področju avtomatizacije in digitalizacije ter za usposabljanje študentov in industrijskih strokovnjakov. Sistem je zasnovan modularno in vključuje pet funkcionalnih proizvodnih celic: skladiščno celico, lasersko gravirno celico, CNC obdelovalno celico, kolaborativno ročno postajo ter celico za vizualno kontrolo kakovosti izdelkov. Celice so povezane s transportnimi sistemi, avtonomnimi mobilnimi roboti in podprte z vrsto senzorjev, ki spremljajo ključne parametre delovanja.

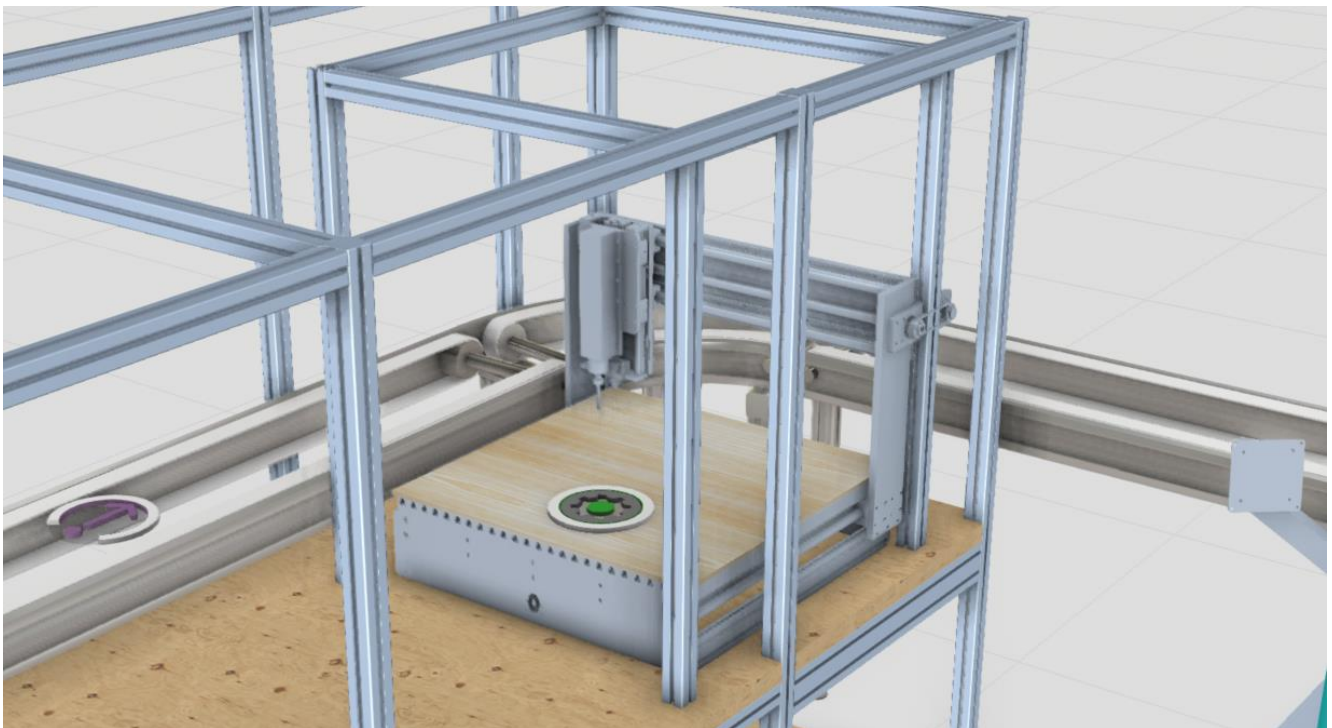
Vsaka od proizvodnih celic ima specifično vlogo v procesu. Skladiščna celica skrbi za pripravo materialov in manipulacijo palet z uporabo industrijskega robota ABB IRB 1100. Laserska gravirna celica omogoča visoko natančno graviranje izdelkov v varnostno zaščitnem okolju. CNC obdelovalna celica združuje robotsko manipulacijo z obdelovalnimi stroji za izvedbo zahtevnih obdelovalnih postopkov. Kolaborativna ročna postaja je zasnovana za sodelovanje človeka in robota Fanuc CRX-10iA, kar omogoča simulacijo sodobnih konceptov sodelovalne proizvodnje. Kontrolna celica pa vključuje vizualni pregled izdelkov z uporabo robota Omron i4-550L in naprednih optičnih sistemov.

### 3 OPIS IZDELAVE DIGITALNEGA DVOJČKA

#### 3.1 MODELIRANJE KOMPONENT

Pri modeliranju komponent smo naleteli na izziv, da mnoge naprave, vključene v sistem, niso bile na voljo v obstoječih knjižnicah Visual Components [1]. Za te naprave smo morali razviti lastne modele [2]. To je vključevalo tudi robotske nadgradnje, kot so vakuumski prijemala za robotske roke in tekoči trak na mobilnem robotu, ter specializirane naprave, kot so laserski gravirni sistemi in CNC obdelovalne enote.

Za vsako komponento smo začeli s preučevanjem tehničnih risb in podatkov o delovnih ciklih. Na podlagi teh informacij smo zasnovali natančne 3D modele kot po meri modeliran CNC stroj, ki ga prikazuje Slika 1. Modeli niso le vizualno predstavljali naprave, ampak so vključevali tudi pomembne animacijske lastnosti, kot so gibanje, omejitve in interakcije z drugimi komponentami. To je omogočilo, da je model deloval v skladu s fizičnimi zakoni, kar je ključnega pomena za realistične simulacije.



Slika 1: Po meri modeliran CNC stroj

### 3.2 POVEZAVA Z REALNIM SISTEMOM PREKO OPC UA

Povezava med digitalnim dvojčkom in realnim sistemom je bila vzpostavljena preko **OPC UA** protokola, ki omogoča standardizirano komunikacijo med različnimi napravami in sistemom za nadzor [3]. Ta protokol omogoča prenos podatkov v realnem času, kar je pomembno za povezovanje simulacije in dejanskega delovanja proizvodnega sistema. Povezovanje virtualnih simulacij s praviimi procesi v realnem času omogoča, da simulacija in dejanska proizvodna linija delujeta sočasno. Digitalni model se tako lahko uporabi za spremljanje, analiziranje in optimizacijo sistemov, pri čemer se podatki iz realnega sveta takoj prenašajo v simulacijo ter obratno.

OPC UA omogoča izmenjavo podatkov med sistemi, pri čemer Visual Components prejema podatke iz fizičnih naprav (npr. senzorjev) ter jih uporablja za poganjanje simulacij. Hkrati lahko simulacije pošiljajo ukaze napravam v realnem času. Podatki se izmenjujejo preko statusnih spremenljivk, ki jih določimo in povežemo preko grafičnega vmesnika, kot to prikazuje Slika 2.

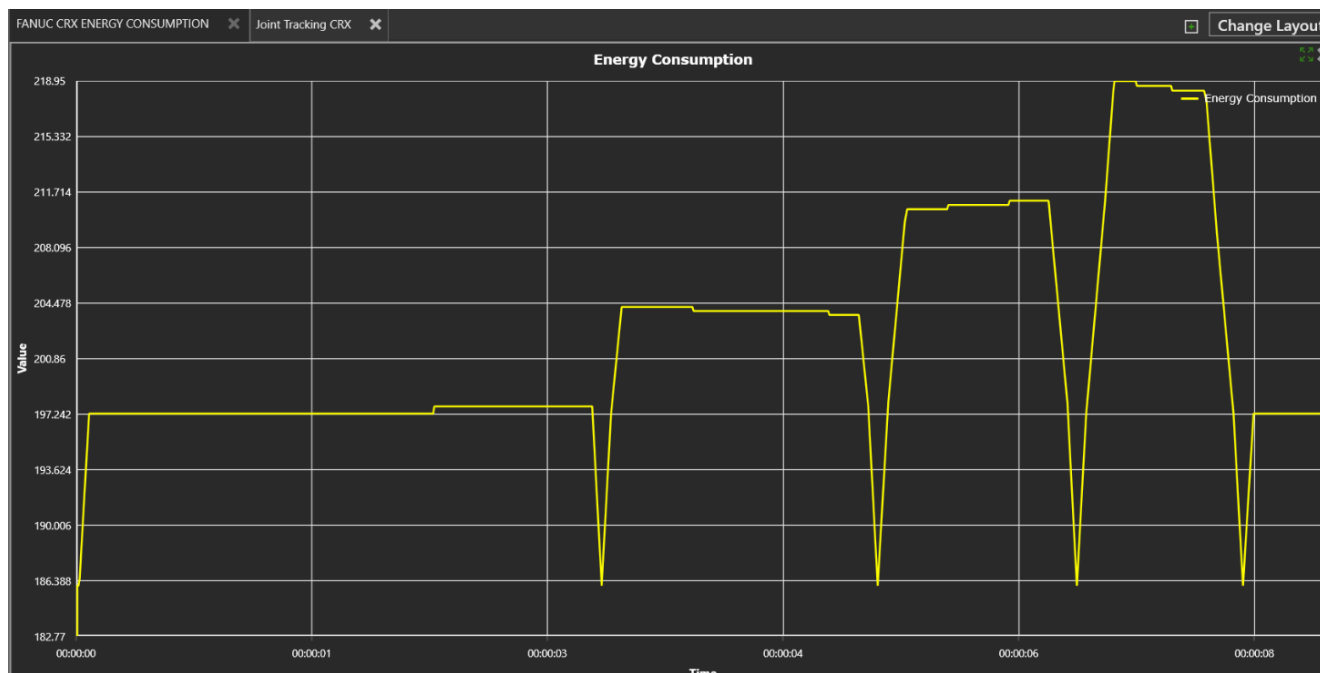
| Connected Variables                                                                                                                                           |                                      |     |             |             |              |                         |                 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|-----|-------------|-------------|--------------|-------------------------|-----------------|
| Structure                                                                                                                                                     | Simulation variable                  | ... | Simulati... | Prepared... | Latest va... | ..                      | Server variable |
|                                                                                                                                                               |                                      |     |             |             |              |                         | Server type     |
| Server                                                                                                                                                        |                                      |     |             |             |              |                         |                 |
| Simulation to server                                                                                                                                          |                                      |     |             |             |              |                         |                 |
| Server to simulation                                                                                                                                          |                                      |     |             |             |              |                         |                 |
| LaserStart                                                                                                                                                    | Process Node.LaserStart              | ⚡   | FALSE       | FALSE       | ✓            | START                   | Boolean         |
| LaserFinish                                                                                                                                                   | Process Node.LaserFinish             | ⚡   | FALSE       | FALSE       | ✓            | STATUS_FINISHED         | Boolean         |
| CNCplace                                                                                                                                                      | Process Node #2.CNCplace             | ⚡   | FALSE       | FALSE       | ✓            | OUT CNC start program   | Boolean         |
| CNCpick                                                                                                                                                       | Process Node #2.CNCpick              | ⚡   | FALSE       | FALSE       | ✓            | OUT CNC end acknowledge | Boolean         |
| VisionStart                                                                                                                                                   | Process Node #5.VisionStart          | ⚡   | FALSE       | FALSE       | ✓            | InSight-Trigger         | Boolean         |
| FanucPick                                                                                                                                                     | From Conveyor Process.FanucPick      | ⚡   | FALSE       | FALSE       | ✓            | PN CRX PNS PROD START   | Boolean         |
| PowerOnSignal                                                                                                                                                 | Conveyor #1.PowerOnSignal            | ⚡   | FALSE       | FALSE       | ✓            | TRAK_VKLOP              | Boolean         |
| PowerOnSignal                                                                                                                                                 | Conveyor #2.PowerOnSignal            | ⚡   | FALSE       | FALSE       | ✓            | TRAK_VKLOP              | Boolean         |
| PowerOnSignal                                                                                                                                                 | Conveyor #3.PowerOnSignal            | ⚡   | FALSE       | FALSE       | ✓            | TRAK_VKLOP              | Boolean         |
| Laser2Start                                                                                                                                                   | Process Node #8.Laser2Start          | ⚡   | FALSE       | FALSE       | ✓            | START                   | Boolean         |
| Laser2Finish                                                                                                                                                  | Process Node #8.Laser2Finish         | ⚡   | FALSE       | FALSE       | ✓            | STATUS_FINISHED         | Boolean         |
| YaskawaPick                                                                                                                                                   | From Conveyor Process #3.YaskawaPick | ⚡   | FALSE       | FALSE       | ✓            | GP 7 PROGRAM RUNNING    | Boolean         |
| Average update time: 101.2 ms    Max update time: 101.3 ms    Pairs with errors: 0<br>Average plugin time: --    Max plugin time: --    Errors on this run: 0 |                                      |     |             |             |              |                         |                 |

Slika 2: Povezava statusnih spremenljivk preko OPC UA strežnika

### 3.3 SISTEM ZA SPREMLJANJE PORABE ENERGIJE

Eden izmed pomembnih delov simulacije, ki smo ga vključili, je bil sistem za spremljanje porabe energije. Ker Visual Components privzeto nima funkcionalnosti za spremljanje porabe energije, smo morali razviti lastne skripte v Pythonu [4], ki so simulirale porabo energije na podlagi delovnih ciklov naprav. Za začetek smo zbrali podatke o običajnih obremenitvah robotov in porabi energije pri različnih stanjih naprave (vklop, pripravljenost, delovanje, stopnje obremenitve). Te podatke smo uporabili za razvoj funkcionalnosti, ki omogoča simulacijo realne porabe energije za robote v laboratoriju. V primeru, da je robot aktiven, se je poraba energije povečala glede na silo s katero je robot deloval ter trajanje premikanje, medtem ko je pri prehodu v stanje pripravljenosti ali izklopa poraba padla na minimum.

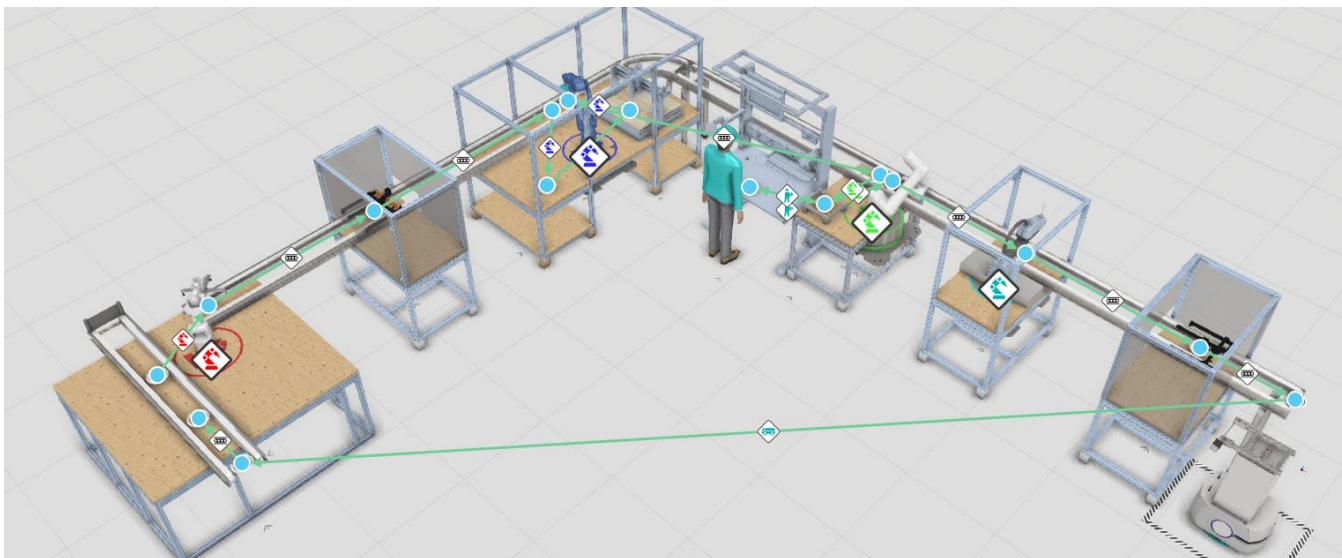
Python skripte so bile integrirane v Visual Components, kar je omogočilo, da so se podatki o porabi energije spremljali in prikazovali v realnem času, ali pa jih tudi izvoziti v excel ali pa .csv datoteko. Poraba energije se je tudi izvajala v Visual Components kot to prikazuje Slika 3. To je omogočilo ne le sledenje porabe energije, temveč tudi analizo različnih scenarijev, kot so optimizacija hitrosti obratovanja, izboljšanje energetske učinkovitosti in preizkušanje različnih energijskih politik, ki bi lahko pripomogle k zmanjšanju stroškov delovanja celotnega sistema.



Slika 3: Graf izmerjene porabe električne energije

### 3.4 POTEK DELA/MODELIRANJE PROCESOV IN ZAGOTAVLJANJE SKLADNOSTI

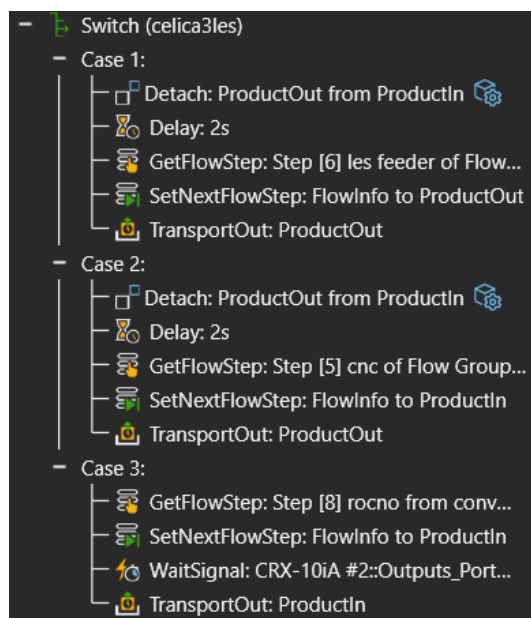
Pri izdelavi poteka dela smo začeli z definiranjem glavnih ciljev, ki so določali smernice za celoten razvoj. Proces je bil zasnovan v več fazah, kjer smo vsako fazo natančno uskladili z zahtevami projekta in potrebami realnega sistema. Celoten postopek se je začel z analizo potrebnih funkcionalnosti in specifikacij za vsako komponento sistema. V prvi fazi smo določili operacije v vsakem delu sistema, od začetne manipulacije materialov v skladiščni celici do končne kontrole kakovosti. Na podlagi teh operacij smo dodali zaporedje nalog za vsak posamezni proces in definirali čase ciklov, ki so določali trajanje posameznih korakov, kot to prikazuje Slika 4. S tem smo pridobili podrobno sliko o tem, kako vsak element deluje znotraj širšega proizvodnega sistema, ter zagotovili, da bodo vsi deli sistema usklajeni [5]. Po tem smo prešli k konfiguraciji procesov, kjer smo nastavili vse parametre, potrebne za delovanje sistema. Vključili smo konfiguracijo posameznih komponent, kot so robotski manipulatorji, CNC stroji in laserski sistemi, ter določili načine njihovega medsebojnega usklajevanja. V tem koraku smo vključili simulacije, ki so nam omogočile testiranje različnih scenarijev obratovanja in preverjanje, kako bi lahko optimizirali proizvodni cikel v različnih pogojih. Z zagotavljanjem skladnosti smo se osredotočili tudi na testiranje in validacijo sistema, kjer smo preverili, ali vse naprave delujejo v skladu z načrtovanimi parametri. Pri tem smo upoštevali dejanske čase obratovanja, hitrosti, časovne zakasnitve ter odzivnost sistema na različne napake. Ta proces je nam je omogočil zagotavljanje, da simulacija ne le ustreza realnemu sistemu, ampak omogoča tudi preizkušanje novih izboljšav in optimizacij brez tveganja za dejanske naprave.



Slika 4: Povezava med procesi v poteku dela

### 3.4.1 DOLOČANJE ŠTEVILA PONOVI TEV CIKLA TER IZBIRO ROBOTSKIH PROGRAMOV TER ZAPOREDJA PROCESOV S POMOČJO SPREMENLJIVK

Znotraj poteka dela smo razvili sistem, ki omogoča dinamično določanje, koliko ponovitev naj se izvede za vsak posamezen cikel, po kakšnem zaporedju naj se posamezni koraki zgodijo in kateri programi naj se izvedejo na različnih robotih ter napravah v proizvodnem procesu. Ta logika, ki jo prikazuje Slika 5 nam omogoča zagotavljanje optimalne sinhronizacije in učinkovitosti sistema, saj omogoča prilagoditev sprememb v proizvodnem toku brez potrebe po dolgotrajni ponovni konfiguraciji. Spremenljivke, ki določajo število ponovitev cikla, zaporedje nalog in specifične programske nastavitve za robote ter druge naprave, se lahko vnesejo na dva načina. Ena možnost je ročni vnos, kjer operater neposredno določi želene parametre, kot so število ciklov, vrstni red nalog in specifične nastavitve za vsako napravo. Druga možnost pa je avtomatski prenos parametrov v logiko simulacije preko OPC UA povezave. Povezava omogoča, da se ti parametri dinamično pošljejo iz realnega sistema v digitalni dvojček, kjer se uporabi za simulacijo različnih proizvodnih scenarijev. Ta pristop zagotavlja, da nismo omejeni na le eno točno določeno zaporedje procesov, temveč omogoča fleksibilnost pri testiranju različnih scenarijev in optimizaciji proizvodnih nalog. Na ta način lahko preizkusimo številne kombinacije in zaporedja nalog, da bi našli najbolj optimalno konfiguracijo za določene cilje, bodisi za povečanje hitrosti proizvodnje, zmanjšanje stroškov, izboljšanje kakovosti ali povečanje energetske učinkovitosti. Fleksibilnost v določanju zaporedja in števila ponovitev ciklov je prednost, saj omogoča, da se sistem hitro prilagodi različnim potrebam, kot so spremembe v obsegu proizvodnje, potrebe po večji prilagodljivosti ali testiranje novih tehnologij v realnem okolju, brez prekinitev obratovanja ali potrebo po temeljitem prestrukturiranju sistema.



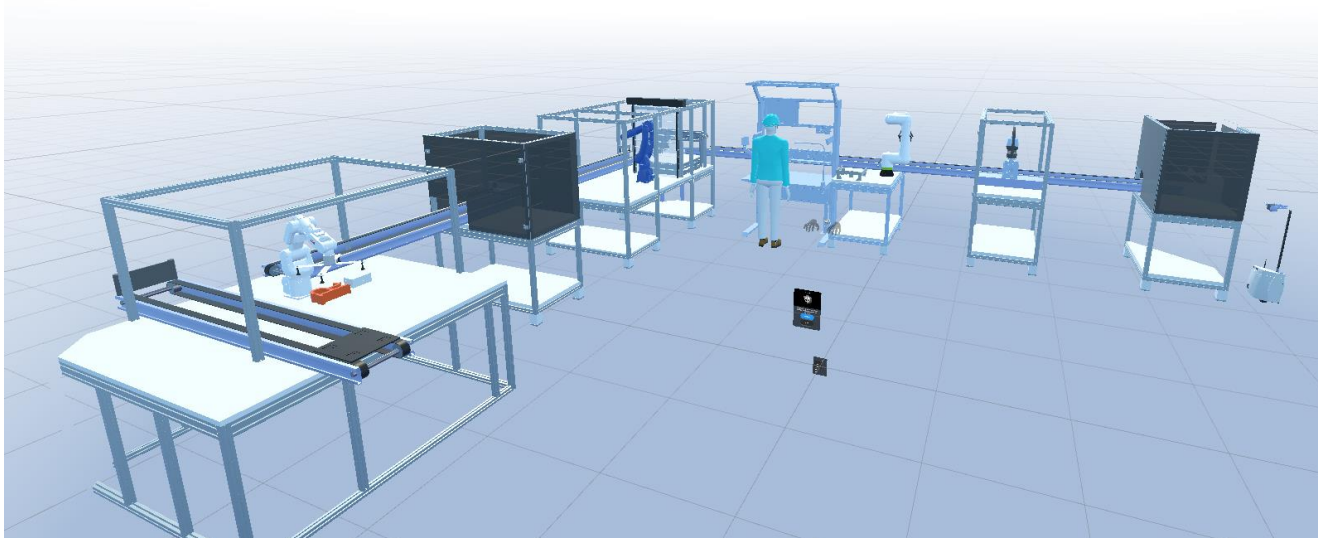
Slika 5: Grafični prikaz programa za določanje naslednjega procesa

## 4 USPOSABLJANJE OPERATERJEV V VIRTUALNEM OKOLJU

Eden od ciljev uporabe digitalnega dvojčka v okviru laboratorijskega sistema LabTOP je bil tudi razvoj orodja za usposabljanje operaterjev v varnem in nadzorovanem virtualnem okolju. V ta namen smo sprva preizkusili uporabo vgrajene funkcionalnosti »Visual Components Experience«, ki omogoča prikaz simulacij v virtualni resničnosti. Kljub osnovni podpori za VR se je pokazalo, da to orodje ne omogoča zadostne interaktivnosti in prilagodljivosti za potrebe usposabljanja, saj uporabniku ne omogoča aktivnega poseganja v potek simulacije ali izvajanja operaterskih nalog znotraj virtualnega prostora.

### 4.1 UVOZ LABORATORIJSKEGA OKOLJA V UNITY

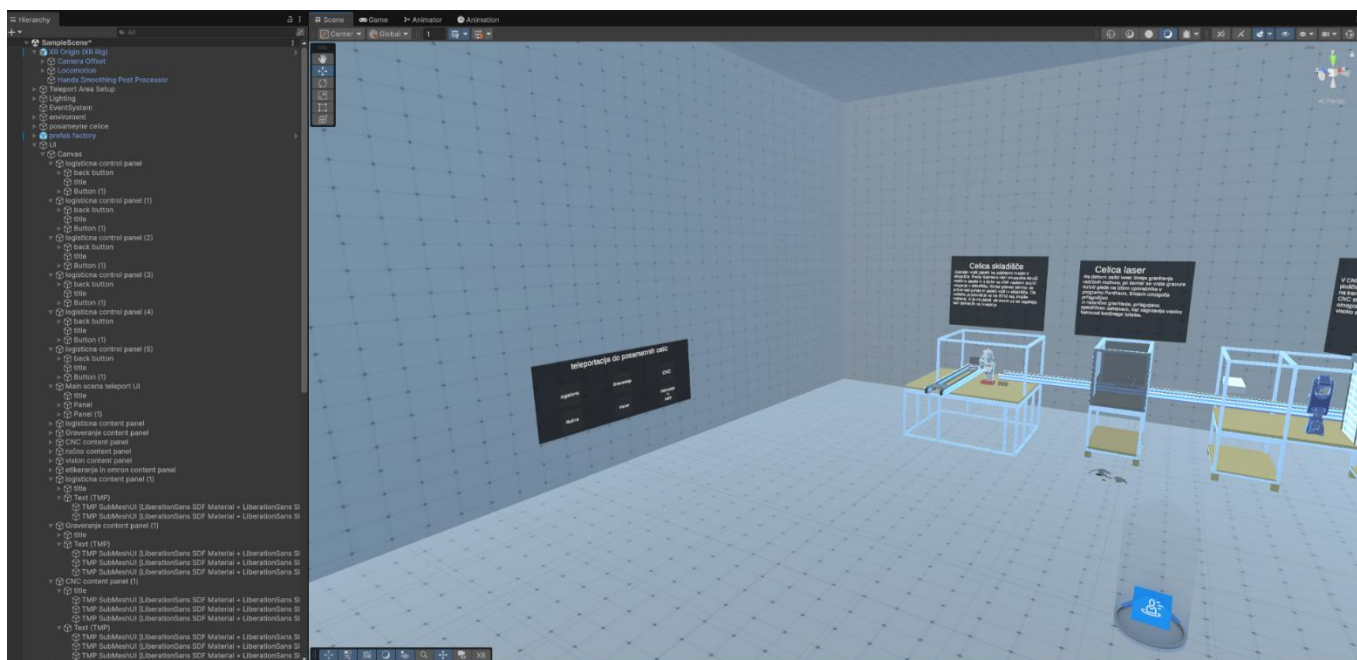
Zaradi omejitev Visual Components Experience modula smo se odločili za alternativni pristop. Celoten 3D model laboratorijskega okolja LabTOP smo izvozili iz okolja Visual Components in ga uvozili v naprednejše razvojno okolje Unity [6]. Na tej osnovi razvijamo interaktivno tridimenzionalno VR okolje, ki omogoča bolj realistično in poglobljeno izkušnjo ter podpira sodelovanje uporabnika. Virtualno usposabljanje v takem okolju uporabnikom omogoča pridobivanje praktičnih znanj brez neposrednega tveganja za poškodbe ali poškodbo opreme. Za prenos 3D-modela LabTOP v Unity smo ga najprej iz okolja Visual Components uvozili v Blender, od tam pa kot Blender-datoteko izvozili v Unity, kar je omogočilo ohranitev vseh animacij.



Slika 6: 3D model v Unityu

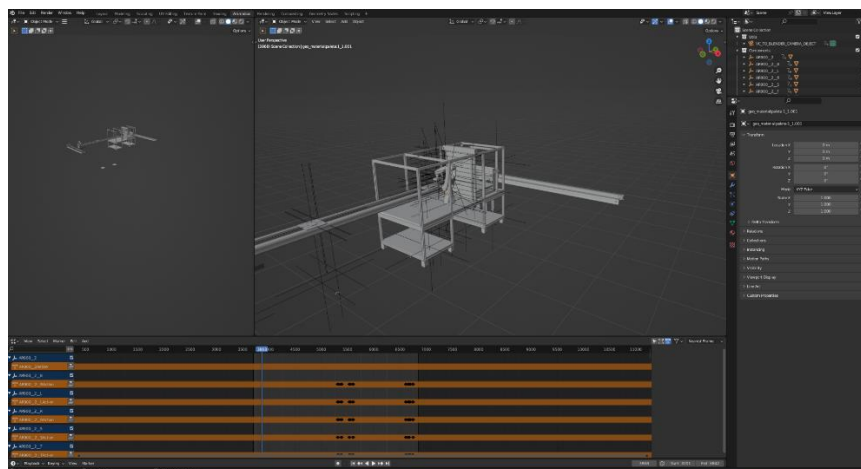
## 4.2 USPOSABLJANJE OPERATERJA NA POSAMEZNI CELICI

Med testiranjem usposabljanja operaterja na laboratorijski liniji LabTOP smo izvedli več simulacijskih scenarijev, v katerih so operaterji izvajali tako celovito kot tudi ciljno usmerjeno delo na posameznih celicah. Ugotovili smo, da se čas, ki ga posvetijo iskanju podrobnih navodil za specifične postopke, znatno podaljša vsakič, ko morajo pregledati celoten potek linije namesto neposrednih navodil za izbrano celico. Da bi sistem prilagodili dejanskim potrebam usposabljanja in izboljšali njegovo učinkovitost, smo v Unity-aplikacijo dodali nov modul za selekcijo celic. Uporabniški vmesnik, ki ga prikazuje Slika 7 sedaj vključuje interaktivni zemljevid linije LabTOP, kjer lahko operater izbere posamezno celico.



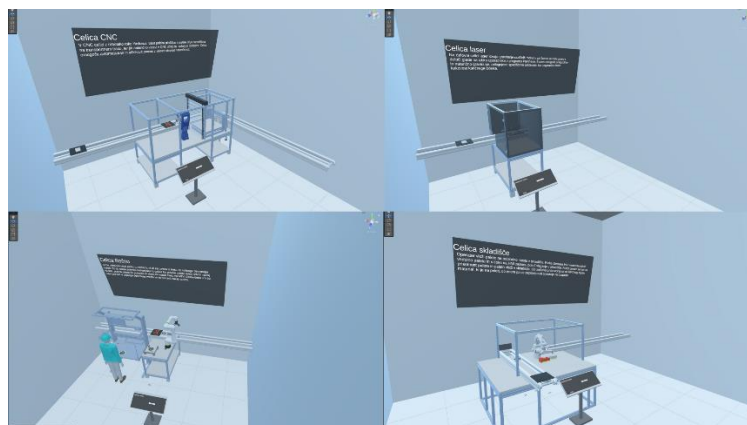
Slika 7: Vmesnik za izbiro postaje

Da bi se animacija proizvodnega poteka izvajala izključno na izbrani celici, smo izvorni 3D-model najprej izvozili iz Unity in ga uvozili v Blender kot .blend datoteko. V Blenderju smo z orodjem Nonlinear Animation [7] razčlenili celotno animacijsko sekvenco na posamezne klipe, pri čemer vsak klip predstavlja eno operacijo znotraj celice kot to prikazuje Slika 8.



**Slika 8: Razrez animacij v programu Blender**

Nato smo posamezne animacije izvozili nazaj v Unity, kjer smo s pomočjo AssetBundle [8] zagotovili dinamično nalaganje ustreznih datotek glede na izbrano celico na interaktivnem zemljevidu. S tem smo dobili posamezne celice na katerih lahko operater opravlja usposabljanje. Slika 9 prikazuje usposabljanje operaterja na posameznih celicah v laboratoriju Labtop.



**Slika 9: Usposabljanje na posameznih celicah**

### 4.3 INTEGRACIJA AI POMOČNIKA V VIRTUALNO OKOLJE UNITY

Za dodatno izboljšanje uporabniške izkušnje pri usposabljanju operaterjev smo v aplikacijo Unity vključili AI-chatbota z uporabo Convai SDK. Convai omogoča hitro povezavo med 3D-avtarjem in velikim jezikovnim modelom (LLM), ki skrbi za naraven pogovor z uporabnikom. Po pregledu razpoložljivih likov smo izbrali avatar Ana, ki služi kot virtualna asistentka v laboratoriju LabTOP (Slika 10).



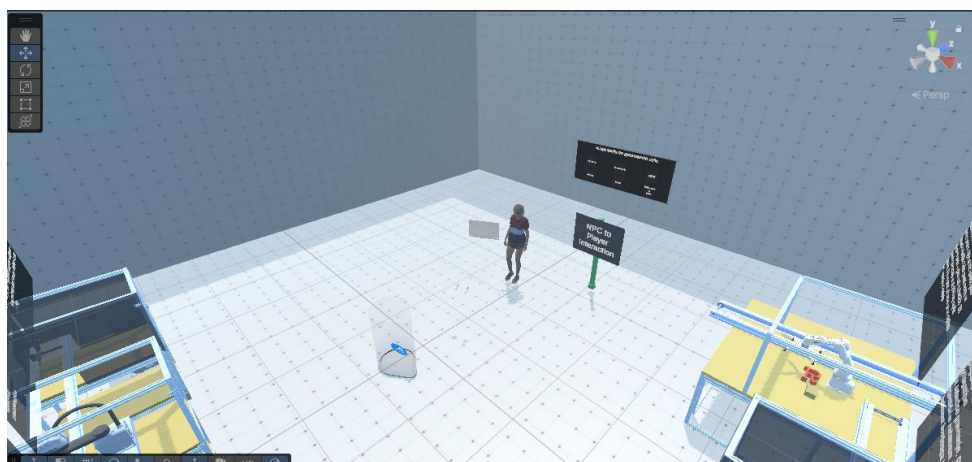
**Slika 10: Virtualni asistent**

#### 4.3.1 Vzpostavitev znanja

Na spletni platformi Convai smo virtualnem asistentu pripravili lastno znanje:

- kratek opis vloge in osebnosti (prijazna in razumljiva asistentka),
- opis vsake posamezne celice na liniji (namen, ključni postopki, varnostni ukrepi),
- splošni opis projekta in prostora (demo linija LabTOP, razporeditev opreme).

S tem pristopom virtualni asistent pozna zgolj informacije, ki smo jih vnesli, kar zagotavlja informacijsko varnost in nadzor nad izhodnimi podatki. Uporabnik lahko z njo komunicira preko mikrofona ali tipkovnice, pri čemer virtualni asistent prepozna bližino določene celice in poda relevantna navodila (Slika 11).



**Slika 11: Tlorisni pogled na postavitev Virtualnega pomočnika**

#### 4.3.2 Tehnični postopek integracije

1. Namestitev Convai SDK v projekt Unity (URP, OpenXR).
2. Dodajanje prefab-komponente ConvaiCharacter in povezava na asistentov ID.
3. Uporaba NavMesh-Agenta za gibanje avatarja med vnaprej določenimi potmi ob liniji.
4. Povezava vhodnih naprav (OpenXR Interaction Toolkit) z ConvaiSpeechController za glasovno ali tekstovno interakcijo.
5. Prilagoditev glasovne povratne informacije z Text-to-Speech (TTS) modulom Convai.

Kot zaključek se je v sklopu aktivnosti v projektu DIGITOP ustvarili digitalni dvojček, ki združuje 3D-model, OPC UA povezavo in nadzor porabe energije. Prehod v Unity z modulom za izbiro celic in AssetBundle animacijami je omogočil bolj interaktivno VR-usposabljanje ter skrajšal čas učenja. Zadnja nadgradnja je virtualni asistent (Convai SDK), ki sledi operaterju, prepozna aktivno celico in glasovno ali pisno poda ciljno usmerjena navodila. S tem se učinkovitost usposabljanja še poveča, prihodnje različice pa bodo z nadzorovanim dostopom do spleta zagotavljale še bolj ažurne informacije.

## 5 NADALJNE IZBOLJŠAVE DIGITALNEGA DVOJČKA

Med implementacijo digitalnega dvojčka smo naleteli na določene pomanjkljivosti programske opreme Visual Components, predvsem na področju optimizacije vrstnega reda naročil, kjer se tipično uporabijo optimizacijski algoritmi. V nadaljevanju bo z obstoječim digitalnim dvojčkom v Visual components integriran še digitalni dvojček v simulacijskem okolju Anylogic, ki nam bo omogočal optimizacijo vrstnega reda naročil in operacij. Združeval bo dodatne module za prediktivno vzdrževanje ter optimizacijo rabe energije, medtem ko bo Visual components uporabljen za natančno simuliranje robotskih procesov.

## BIBLIOGRAFIJA

- [1] „Introducing Visual Components Experience 1.7“, Visual Components. Pridobljeno: 10. julij 2025. [Na spletu]. Dostopno na: <https://www.visualcomponents.com/blog/introducing-visual-components-experience-1-7/>
- [2] „Basics of Component Modeling“, Visual Components Academy. Pridobljeno: 10. julij 2025. [Na spletu]. Dostopno na: <https://academy.visualcomponents.com/courses/basics-of-component-modeling/>
- [3] „OPC UA Server - General Questions“, Visual Components - The Simulation Community. Pridobljeno: 10. julij 2025. [Na spletu]. Dostopno na: <https://forum.visualcomponents.com/t/opc-ua-server/1931>
- [4] „Blender Addon - Blenderer - Extensions and Python Add-ons / Visual Components Add-on Samples“, Visual Components - The Simulation Community. Pridobljeno: 10. julij 2025. [Na spletu]. Dostopno na: <https://forum.visualcomponents.com/t/blender-addon-blenderer/773>
- [5] „Process Modeling Flows“, Visual Components Academy. Pridobljeno: 10. julij 2025. [Na spletu]. Dostopno na: <https://academy.visualcomponents.com/courses/process-modeling-flows/>
- [6] „Introduction to Digital Twins with Unity“, Unity Learn. Pridobljeno: 10. julij 2025. [Na spletu]. Dostopno na: <https://learn.unity.com/tutorial/https://learn.unity.com/tutorial/introduction-to-digital-twins-with-unity>
- [7] „Nonlinear Animation - Blender 4.4 Manual“. Pridobljeno: 10. julij 2025. [Na spletu]. Dostopno na: <https://docs.blender.org/manual/en/latest/editors/nla/index.html>
- [8] U. Technologies, „Unity - Manual: Using AssetBundles Natively“. Pridobljeno: 10. julij 2025. [Na spletu]. Dostopno na: <https://docs.unity3d.com/2022.1/Documentation/Manual/AssetBundles-Native.html>