

Program

Digitalna transformacija robotiziranih tovarn prihodnosti

DIGITOP

Poročilo D2.2

Algoritmi optimalne rekonfiguracije

Projekt/RRP
Vrsta dokumenta
Verzija dokumenta
Sodelujoči partnerji
Avtorji
Datum priprave

D2.2
Poročilo o rezultatih
Končna
IJS-E1
Miha Deniša, Jaka Jereb
4.11.2025

Kazalo

1	Uvod	3
2	Pregled stanja in izzivi vizualne kontrole kakovosti	3
3	Sistem vizualne kontrole	4
3.1	Kinematična zgradba in prostor stanj	4
3.2	Čas prehoda med stanji	6
3.3	Predstavitev sistema z grafi	6
4	Optimizacija postavitve kamer	7
4.1	Matematična formulacija problema	7
4.2	Omejitve optimizacijskega problema	7
4.3	Postopek reševanja in numerična optimizacija	8
5	Optimizacija zaporedja zajemov	8
5.1	Metode reševanja in rezultati	8
5.2	Dvo-kamerna konfiguracija in planarno grupiranje	9
5.3	Povezava z optimizacijo postavitve kamer	10
6	Fleksibilni pasivni monopod kot orodje za optimalno rekonfiguracijo	10
7	Nadgradnje teoretičnih modelov in povezava z nadaljnjim razvojem	11
	Literatura	12

1 Uvod

Program **DIGITOP – Digitalna transformacija robotiziranih tovarn prihodnosti** naslavlja ključne izzive sodobne industrije na poti k digitalnemu in trajnostnemu prehodu. V okviru programa poteka več medsebojno povezanih raziskovalno–razvojnih projektov (RRp), ki skupaj tvorijo celovito verigo od temeljnih raziskav do eksperimentalne validacije in industrijskih demonstracij. Poročilo D2.2 je rezultat aktivnosti v okviru **RRp2: Industrijske raziskave – Rekonfiguracija, modularnost in avtonomna prilagoditev za napredne robotske celice**.

Namen poročila D2.2 je predstaviti algoritme optimalne rekonfiguracije, ki omogočajo samodejno izračunavanje optimalne zasnove rekonfigurabilne, modularne robotske delovne celice ter izbor ustreznega orodja in senzorike. Cilj razvitega pristopa je zagotoviti, da se konfiguracija celice – položaji robotov, prijemal, kamer in drugih modulov – lahko samodejno prilagodi novi nalogi z minimalnim časom nastavitve in največjo učinkovitostjo delovanja. Optimizacija rekonfiguracije se izvaja na osnovi digitalnih modelov celice in omogoča iterativno izboljševanje postavitve komponent ter zaporedja robotskih operacij.

Raziskave predstavljene v tem poročilu temeljijo na kombinaciji optimizacijskih pristopov, kinematičnega modeliranja in analize gibanja robota, ki omogočajo matematično formulacijo problema zasnove in optimizacije konfiguracije celice. Predstavljeni algoritmi obravnavajo dva ključna vidika rekonfiguracije: (1) optimizacijo postavitve komponent delovne celice, zlasti senzorjev in kamer, glede na vsak posamezni obdelovanec, ter (2) optimizacijo zaporedja robotskih operacij oziroma preglednih točk, z namenom minimizacije časa cikla in povečanja učinkovitosti.

Pri razvoju algoritmov se opiramo na rezultate, objavljene v konferenčnih prispevkih *“Optimizing Camera Placement in Agile Robotic Cells for Visual Inspection”* [3] in *“Sequence Optimization in Multi-Camera Robotic Visual Inspection”* [4], ki obravnavata optimizacijo postavitve kamer in zaporedja zajemov slik v okviru rekonfigurabilnih preglednih celic. V okviru RRp2 so ti pristopi nadgrajeni in prilagojeni širšemu problemu optimalne rekonfiguracije, ki vključuje tudi modularno zasnovo celice in izbiro orodij.

Poročilo je strukturirano tako, da najprej predstavi pregled stanja na področju optimizacije konfiguracije robotskih celic in vizualne kontrole kakovosti, nato pa obravnava matematično modeliranje, optimizacijo postavitve kamer in zaporedja operacij. V zaključku so podane usmeritve za vključitev razvitih algoritmov v digitalne dvojčke in nadaljnji eksperimentalni razvoj v okviru **RRp6**.

2 Pregled stanja in izzivi vizualne kontrole kakovosti

Vizualna kontrola kakovosti je eno ključnih področij avtomatizirane proizvodnje, saj neposredno vpliva na zanesljivost, učinkovitost in sledljivost procesov. Z razvojem konceptov *Industrije 4.0* ter prehodom k digitalnim in rekonfigurabilnim tovarnam prihodnosti naraščajo zahteve po inteligentnih preglednih sistemih, ki se lahko samodejno prilagajajo različnim izdelkom, orodjem in delovnim pogojem [8]. Pri tem klasične, statično konfigurirane celice, zasnovane za serijsko proizvodnjo, niso več primerne za okolja z visoko variabilnostjo (*high-mix, low-volume*), kjer se konfiguracija proizvodne linije pogosto spreminja. Zato se raziskave usmerjajo v razvoj fleksibilnih robotskih celic z vgrajenimi metodami optimizacije, samoučenja in inteligentnega načrtovanja gibanja [6].

V takšnih agilnih celicah ima vizualni sistem razširjeno vlogo: poleg osnovne kontrole kakovosti omogoča tudi samodejno poravnavo izdelkov, sledenje procesom in podporo človeku pri montaži ali validaciji. Ključno vprašanje pri zasnovi takih sistemov je, kako postaviti kamere in robota tako, da zagotovimo popolno pokritost pregledanih površin ob minimalnem času cikla. Pravilna izbira pozicij kamer in zaporedja gibov robota zmanjša trajanje pregleda, poveča natančnost in izboljša zanesljivost zaznave napak.

V okviru raziskav predstavljenih v [3] je bila razvita metoda za optimizacijo postavitve kamer v rekonfigurabilni pregledni celici, kjer robot drži izdelek med pregledom. Sistem je bil modeliran z uporabo homogenih transformacijskih matrik in prostora stanj, v katerem vsako stanje predstavlja kombinacijo kamere in pregledne točke. Ciljna funkcija je bila definirana kot skupni čas trajanja pregleda, ki se minimizira z iterativnim nelinearnim programiranjem ob upoštevanju kinematike robota, geometrije kosa in vidnega polja kamer. Rezultati eksperimentalne validacije so pokazali, da optimalno izbrana postavitve kamer omogoča pomembno skrajšanje časa cikla brez zmanjšanja pokritosti površin. Podobne zasnove postavitve senzorjev so uporabljali tudi drugi avtorji, npr. Erdem in Sclaroff [5] pri avtomatskem načrtovanju kamer v prostorih ter Olague in Mohr [10] pri določanju optimalnih pogledov za 3D rekonstrukcijo.

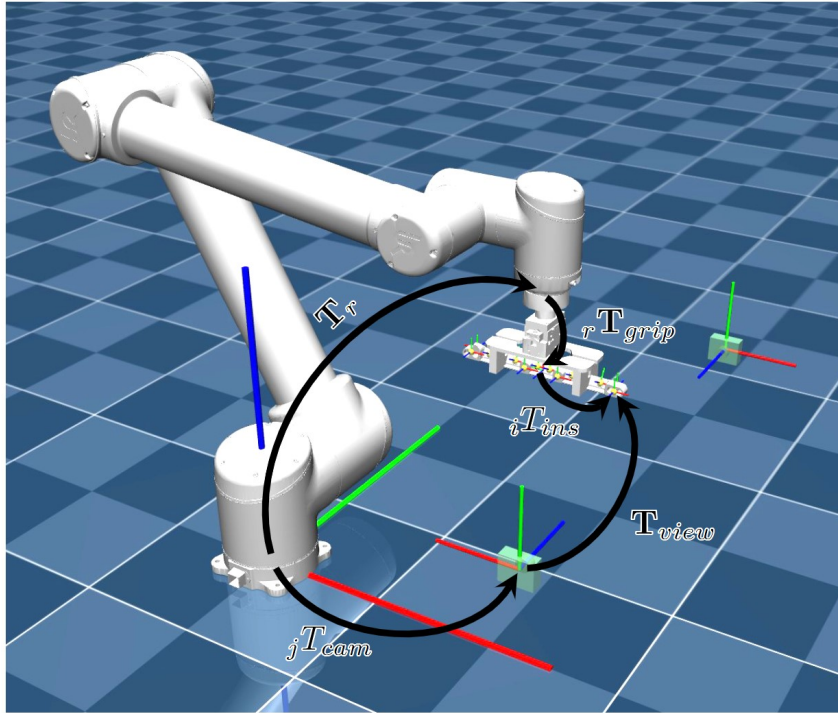
Drugi ključni problem vizualne kontrole je določitev optimalnega zaporedja zajemov slik, ki neposredno vpliva na učinkovitost delovanja sistema. V prispevku [4] je bil ta problem obravnavan kot kombinatorični optimizacijski problem, pri katerem so vozlišča grafa predstavljala pregledne točke, uteži med njimi pa čase prehodov robota, izračunane na podlagi največjega premika sklepov med konfiguracijami. Za reševanje so bile uporabljene tri metode: heuristična 2-opt [1], binarno celoštevilsko programiranje (BIP) [11] in dinamično programiranje Held–Karp [7]. Rezultati so pokazali, da 2-opt doseže skoraj optimalne rešitve (znotraj 1–2 % od optima) z zanemarljivim računskim časom, kar omogoča uporabo v realnih preglednih procesih. Eksaktni pristopi, kot sta BIP in Held–Karp, pa služijo kot referenčna osnova za validacijo kakovosti rešitev in oceno optimalnosti. Posebnost dvo-kamerne postavitve je bila uvedba pristopa *planarnega grupiranja*, pri katerem se pregledne točke na isti ravnini dodelijo posamezni kameri. Z uporabo več kamer se zmanjšajo potrebni premiki robota, kar omogoči vzporedno delovanje obeh kamer. V simulacijah je to privedlo do 1.8–2.3-krat krajšega časa cikla v primerjavi z enokamernim sistemom.

3 Sistem vizualne kontrole

Za učinkovito optimizacijo postopkov vizualne kontrole je potrebno najprej vzpostaviti matematični model sistema, ki opisuje prostorske odnose med robotom, kamerami in pregledovanimi točkami. Tak model omogoča analitično oceno vidnosti, dosegljivosti in časovne zahtevnosti posameznih pregledov, ter služi kot osnova za nadaljnjo optimizacijo postavitve kamer in zaporedja zajemov.

3.1 Kinematična zgradba in prostor stanj

Model obravnava robotsko celico, v kateri manipulator drži izdelek, medtem ko so kamere nameščene v fiksni položajih. Simulirana celica za vizualni pregled z kinematično verigo je prikazana na sliki 1. Za opis sistema uporabimo homogene transformacijske matrike, ki povezujejo različne koordinatne sisteme: vrh robota (T_r), relativno pozo točke prijema (${}_rT_{grip}$), kamero (${}_jT_{cam}$), vidni



Slika 1: Simulirana celica za vizualni pregled z kinematično verigo.

položaj kamere (T_{view}) in pregledno točko na izdelku (iT_{ins}). Celotno kinematično verigo lahko zapišemo kot:

$$\mathbf{T}_r = {}_j\mathbf{T}_{cam} \mathbf{T}_{view} {}_i\mathbf{T}_{ins}^{-1} {}_r\mathbf{T}_{grip}^{-1} \quad (1)$$

kjer $j = 1, \dots, n$ označuje kamero in $i = 1, \dots, m$ posamezno pregledno točko. Vsak par (i, j) torej določa specifično konfiguracijo robota, pri kateri je točka i v vidnem polju kamere j .

Za reševanje in optimizacijo gibanja je potrebno izračunati sklepe robota, ki ustrezajo dani poziciji končnega člena. Z uporabo funkcije inverzne kinematike $f_{IK}(\cdot)$ se konfiguracija robota določi kot:

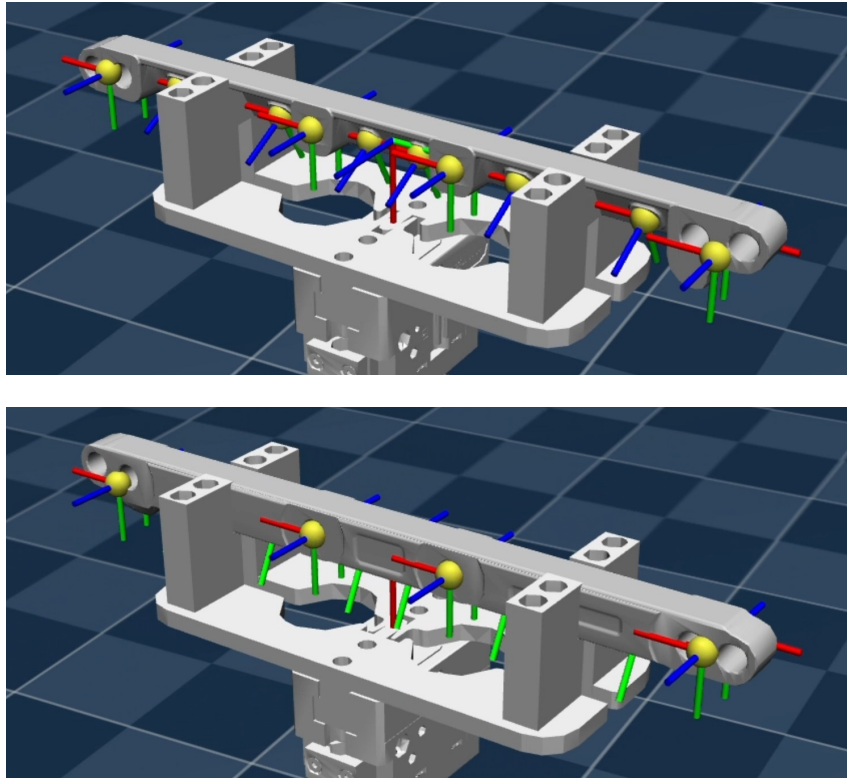
$$q_r^{(i,j)} = f_{IK}(T_r^{(i,j)}), \quad (2)$$

kjer $q_r^{(i,j)}$ predstavlja vektor sklepov robota za posamezno kombinacijo kamere in pregledne točke.

Prostor možnih konfiguracij definiramo kot množico vseh stanj $s(i, j)$, pri čemer vsako stanje opisuje specifično kombinacijo kamere in točke, ki jo ta kamera zajema:

$$S = \{s(i, j) \mid i \in \{1, \dots, m\}, j \in \{1, \dots, n\}\}. \quad (3)$$

Vsak prehod med dvema stanjem $s_1, s_2 \in S$ je ovrednoten s časom, potrebnim za premik robota med njima.



Slika 2: Prikaz robotskega prijemala, ki drži industrijski predmet za vizualni pregled. Podana sta dva pogleda. Pregledne točke (IP) so označene z rumenimi krogli in pripadajočimi koordinatnimi sistemi. Opazimo lahko, da štirinajst preglednih točk leži v treh različnih ravninah.

3.2 Čas prehoda med stanji

Čas, ki ga robot potrebuje za premik med dvema konfiguracijama q_1 in q_2 , je odvisna od največjega premika posameznega sklepa. Če predpostavimo enake maksimalne hitrosti $\dot{q}_{max}$ in pospeške $\ddot{q}_{max}$ za vse sklepe, je trajanje prehoda med stanji s_1 in s_2 določeno kot [4]:

$$d(s_1, s_2) = \frac{\dot{q}_{max}}{\ddot{q}_{max}} + \frac{\hat{q}_{max}^{(s_1, s_2)}}{\dot{q}_{max}} + \hat{q}_{add}, \quad (4)$$

kjer $\hat{q}_{max}^{(s_1, s_2)}$ označuje največji sklepni premik med konfiguracijama, $\hat{q}_{add}$ pa zajema dodatne faktorje (čas sprožitve kamere, stabilizacijo slike ipd.). Tak model omogoča oceno časovnih stroškov za vse prehode med pari stanj, kar je osnova za formulacijo problema optimizacije zaporedja zajemov v obliki grafa.

3.3 Predstavitev sistema z grafi

Sistem lahko na podlagi definirane prostora stanj predstavimo kot popolnoma povezan, usmerjen in utežen graf [1]:

$$G = (S, E, w), \quad (5)$$

kjer S predstavlja množico vseh stanj (vozlišč), E množico robov (prehodi med stanji), utež $w(s_1, s_2)$ pa predstavlja čas prehoda med danima konfiguracijama, določen z enačbo (4). Na tej osnovi je mogoče definirati optimizacijske probleme, ki iščejo najkrajšo pot skozi graf, pri čemer se vsaka pregledna točka obišče natanko enkrat. Takšna formulacija omogoča uporabo standardnih algoritmov za optimizacijo poti, kot so 2-opt heuristika ali BIP pristopi, kar bo podrobneje obravnavano v naslednjih poglavjih.

4 Optimizacija postavitve kamer

Eden ključnih ciljev pri načrtovanju rekonfigurabilnih robotskih celic je določitev takšnih pozicij kamer, ki omogočajo popolno pokritost preglednih točk z minimalnim skupnim časom pregleda. Postavitev kamer vpliva na dolžino poti robota, število potrebnih premikov in kakovost zajetih slik, zato predstavlja osrednji optimizacijski problem pri načrtovanju preglednih sistemov v modularnih celicah.

4.1 Matematična formulacija problema

Vsaka poza je določena s translacijskim vektorjem $\mathbf{t} = [x, y, z]^T$ in rotacijskim delom, opisanim z rotacijsko matriko $R \in SO(3)$. Razdalja med pozama je opredeljena kot vsota translacijskega in rotacijskega odklona [3]:

$$\Upsilon(p_1, p_2) = \|\mathbf{t}_1 - \mathbf{t}_2\|^2 + \lambda_r \|\log(R_1 R_2^T)\|^2, \quad (6)$$

kjer parameter λ_r uravnava relativni vpliv orientacije glede na premik. Rotacijska razdalja se izrazi z logaritmom produkta rotacijskih matrik, kar omogoča opis minimalnega vrtenja med dvema orientacijama v prostoru $SO(3)$.

Za vsak nabor možnih poz kamer $p_c = [p_1, \dots, p_n]^T$ definiramo funkcijo cilja, ki meri skupni strošek postavitve. Funkcija je sestavljena iz dveh členov: (1) penalizacije odmika od začetne lege kamer in (2) minimalnega časa pregleda za dano postavitve [3]:

$$c(p_c) = \lambda_t \sum_{j=1}^n \Upsilon(p_j, p_{j,0}) + d_{\min}(p_c), \quad (7)$$

kjer $\Upsilon(\cdot)$ označuje razdaljo med pozami, definirano v enačbi (6), $p_{j,0}$ pa začetno ali referenčno lego kamere j . Prvi člen v funkciji cilja deluje kot regularizacija, ki preprečuje nerealne premike kamer, drugi člen $d_{\min}(p_c)$ pa predstavlja najkrajši možni skupni čas pregleda za dano konfiguracijo kamer. Ta čas se določa z reševanjem problema optimalnega zaporedja zajemov, kot je opisano v poglavju 5.

4.2 Omejitve optimizacijskega problema

Problem optimizacije postavitve kamer je dodatno omejen s fizikalnimi in geometrijskimi pogoji, ki zagotavljajo izvedljivost rešitev. Za vsako kamero j so definirane spodnje in zgornje meje položaja in orientacije:

$$\mathbf{p}_j^{\min} \leq \mathbf{p}_j \leq \mathbf{p}_j^{\max}, \quad i = j, \dots, n, \quad (8)$$

pri čemer $\mathbf{p}_j = [x_j, y_j, z_j, \alpha_j, \beta_j, \gamma_j]^T$ predstavlja vektor translacijskih in rotacijskih komponent lege kamere. Omejitve zagotavljajo, da kamere ostanejo znotraj določenega delovnega prostora in da njihove orientacije ne povzročijo okluzij ali kolizij.

4.3 Postopek reševanja in numerična optimizacija

Optimizacija funkcije cilja (7) je bila izvedena kot nelinearni iterativni postopek, v katerem se lege kamer postopoma prilagajajo, dokler funkcija ne doseže stacionarne vrednosti [3]. V vsaki iteraciji optimizacije je bilo potrebno za trenutno konfiguracijo kamer izračunati minimalni čas pregleda $d_{\min}(p_c)$. Ta vrednost se določi z reševanjem problema *potujočega trgovca* (TSP), ki opisuje optimalno zaporedje zajemov slik glede na razdalje med preglednimi točkami, kot je podrobno opisano v [4].

Za vsako kombinacijo kamer in preglednih točk je bila vnaprej izračunana matrika časov prehoda med posameznimi konfiguracijami robota, določena po enačbi (4). Na osnovi teh uteži se rešuje TSP, katere rezultat je zaporedje, ki minimizira skupni čas trajanja pregleda. Izračunani čas predstavlja minimalni možni čas cikla za dano postavitev kamer, kar omogoča posodobitev funkcije cilja (7).

Optimizacija se zaključi, ko spremembe vrednosti funkcije cilja postanejo manjše od praga konvergence. Na ta način se določi optimalna konfiguracija kamer p_c^* in pripadajoče optimalno zaporedje preglednih točk, ki skupaj minimizirata skupni čas pregleda ob ohranjeni popolni pokritosti opazovanih površin.

5 Optimizacija zaporedja zajemov

Določitev optimalnega zaporedja zajemov je ključna za izračun minimalnega časa pregleda $d_{\min}(p_c)$ pri dani postavitvi kamer. Problem je bil v [4] modeliran kot *problem potujočega trgovca* (TSP), kjer posamezna pregledna točka predstavlja vozlišče grafa, uteži med točkami pa predstavljajo čas prehoda robota med konfiguracijami. Cilj optimizacije je minimizirati skupni čas gibanja robota med vsemi točkami pregleda:

$$\min_{\pi \in \Pi} \sum_{k=1}^{|S|-1} w(s_{\pi_k}, s_{\pi_{k+1}}), \quad (9)$$

pri čemer Π predstavlja množico vseh možnih permutacij vozlišč, S pa množico vseh stanj, določenih z različnimi kombinacijami kamer in preglednih točk. Tako dobljeno zaporedje se nato uporabi v postopku optimizacije postavitve kamer za izračun $d_{\min}(p_c)$.

5.1 Metode reševanja in rezultati

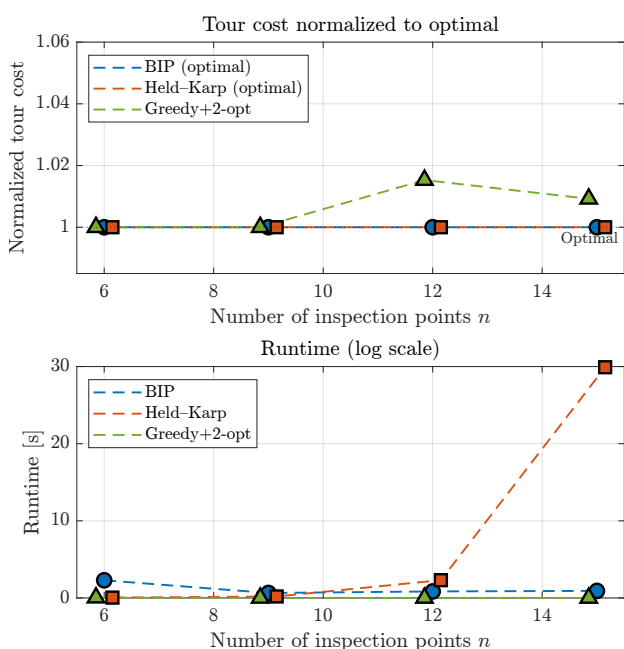
V [4] so bili analizirani trije pristopi reševanja TSP: (1) heuristična metoda 2-opt, (2) eksaktno binarno celoštevilsko programiranje (BIP), in (3) dinamično programiranje Held–Karp. Rezultati so pokazali, da 2-opt zagotavlja skoraj optimalne rešitve (znotraj 1–2 % od optimuma) ob bistveno

krajšem času računanja, kar omogoča uporabo v iterativnem postopku optimizacije postavitve kamer. BIP in Held–Karp pristopa sta služila kot referenčni metodi za vrednotenje kakovosti heurističnih rešitev.

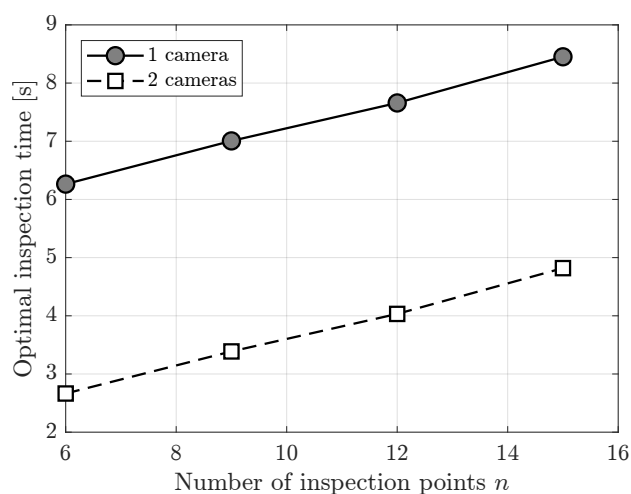
Na sliki 3 je prikazana primerjava rezultatov za različne velikosti problemov in konfiguracije kamer. Zgornji del prikazuje normalizirane vrednosti dolžin poti za vse konfiguracije kamer in preglednih ravnin, spodnji del pa pripadajoče čase računanja na logaritemski skali. Metoda 2-opt ohranja odstopanje manjše od 1.3% od optimalne rešitve pri vseh testih in izkazuje skoraj konstantno časovno zahtevnost, kar jo naredi primerno za vgradnjo v iterativni optimizacijski okvir.

5.2 Dvo-kamerna konfiguracija in planarno grupiranje

V dvo-kamerni konfiguraciji je bila uporabljena strategija *planarnega grupiranja*, pri kateri se pregledne točke razvrstijo po ravninah objekta in dodelijo posameznim kameram. S tem se omogoči delno vzporedno izvajanje pregledov.



Slika 3: Primerjava metod reševanja TSP za dvo-kamerno konfiguracijo z razvrstitvijo preglednih ravnin po skupinah [4]. Metoda 2-opt ostaja znotraj 1.3 % od optimuma.



Slika 4: Primerjava optimalnih časov pregledov (BIP) za eno- in dvo-kamerno konfiguracijo [4]. Izboljšanje je izraženo kot razmerje $T_{1\text{ cam}}/T_{2\text{ cam}}$.

Rezultati prikazujejo, da optimizirana dvo-kamerna konfiguracija omogoča do 2,3-kratno izboljšanje časa cikla v primerjavi z enokamernim sistemom, pri čemer se ohranja popolna pokritost pregledanih površin. S tem se potrjuje učinkovitost strategije združevanja preglednih točk po ravninah in vzporednega zajemanja slik.

5.3 Povezava z optimizacijo postavitve kamer

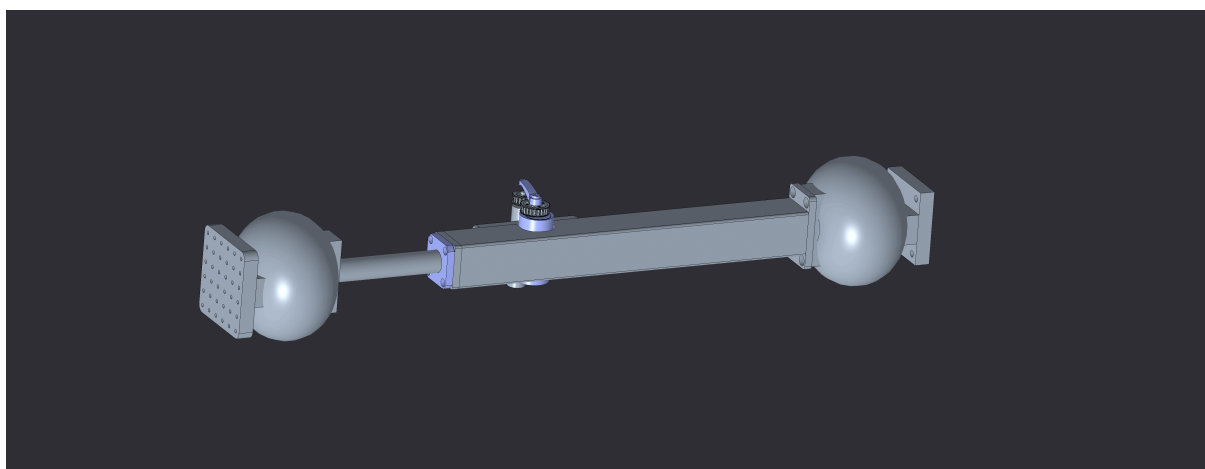
Optimizirano zaporedje zajemov neposredno določa minimalni čas pregleda $d_{\min}(p_c)$, ki se uporablja v funkciji cilja pri optimizaciji postavitve kamer (enačba 7). S tem je zagotovljena povezanost obeh nivojev optimizacije: prostorska postavitev kamer določa možne konfiguracije preglednih točk, optimizacija zaporedja pa določa najkrajši možni čas cikla za dano postavitev. Tako združena rešitev predstavlja temelj za samodejno rekonfiguracijo in načrtovanje učinkovite vizualne kontrole v modularnih robotskih celicah.

6 Fleksibilni pasivni monopod kot orodje za optimalno rekonfiguracijo

Pomemben del raziskav, povezanih z razvojem algoritmov za samodejno rekonfiguracijo modularnih robotskih celic, je razvoj novega mehanskega elementa — fleksibilnega pasivnega monopoda. Monopod predstavlja enostavnejšo, stroškovno učinkovito različico fleksibilnega pasivnega heksapoda, ki je bil v preteklosti uporabljen kot prilagodljiv vpenjalni sistem v laboratorijskih in industrijskih okoljih.

Monopod temelji na uporabi ene noge heksapoda [9, 2], kar mu omogoča večje delovno območje pri bistveno manjši kompleksnosti. Gre za mehanski nosilni sistem brez lastnih aktuatorjev, ki ga je mogoče prestavljati z robotsko roko in ga po dosegu želenega položaja fiksirati z vgrajenimi zavorami. Na ta način se doseže togost in ponovljivost primerljiva z aktivnimi sistemi, vendar z bistveno nižjimi stroški in brez potrebe po dodatnem napajanju ali krmiljenju.

V okviru projekta DIGITOP bo monopod služil kot nosilec industrijske kamere za potrebe vizualne kontrole kakovosti. Njegova zasnova omogoča, da robot kamero namesti v optimalno lego glede na trenutno konfiguracijo delovne celice, pri čemer algoritmi, opisani v tem poročilu, določijo optimalno lego in orientacijo. S tem monopod predstavlja fizično realizacijo spremenljive konfiguracije senzorjev, kar neposredno podpira cilj D2.2 – razvoj algoritmov optimalne rekonfiguracije delovne celice in zahtevanega orodja.



Slika 5: Fleksibilni pasivni monopod (ilustrativni prikaz). Robot ga lahko premešča med različne lege, kjer služi kot nosilec kamere ali drugega senzorja.

Integracija monopoda v rekonfigurabilno celico omogoča fizično realizacijo rezultatov optimizacijskih algoritmov, razvitih v tem poročilu, in predstavlja ključen korak k popolnoma prilagodljivim modularnim sistemom. S tem se povezuje teoretični razvoj algoritmov v okviru RRP2 z njihovo eksperimentalno validacijo, ki bo izvedena v projektu RRP6 in predstavljena v poročilu D6.1.

7 Nadgradnje teoretičnih modelov in povezava z nadaljnjim razvojem

Raziskave, predstavljene v tem poročilu, združujejo dva komplementarna pristopa k optimizaciji rekonfigurabilnih robotskih celic: (1) optimizacijo postavitve kamer glede na konfiguracijo robota [3] in (2) optimizacijo zaporedja preglednih točk z uporabo formulacije problema potujočega trgovca [4]. Oba pristopa tvorita celovit optimizacijski okvir za samodejno načrtovanje in rekonfiguracijo modularnih robotskih sistemov.

Razvite metode predstavljajo temelj za nadgradnjo modelov, ki bodo v nadaljevanju projekta DIGITOP uporabljeni pri eksperimentalni validaciji v okviru **RRp6**. Poseben poudarek bo namenjen integraciji algoritmov v programsko okolje ROS in uporabniški vmesnik (MAR), kar bo omogočilo neposredno uporabo razvite programske knjižnice za optimizacijo v rekonfigurabilni proizvodni celici.

Hkrati bodo algoritmi podprli fizično rekonfiguracijo senzorike s pomočjo fleksibilnega pasivnega monopoda, ki bo služil kot nosilec kamere ali drugega orodja. S tem bo omogočeno eksperimentalno preverjanje matematičnih modelov in njihovih napovedi v realnem delovnem okolju. Tako se bodo rezultati D2.2, razviti na raziskovalni ravni (TRL 4), prenesti v eksperimentalno validacijo in demonstracijo na ravni TRL 6.

Literatura

- [1] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2014.
- [2] M. Bem, M. Deniša, T. Gašpar, J. Jereb, R. Bevec, I. Kovač, and A. Ude. Reconfigurable fixture evaluation for use in automotive light assembly. In *Proc. 18th Int. Conf. Adv. Robot. (ICAR)*, pages 61–67, Hong Kong, China, 2017.
- [3] M. Deniša, T. Gašpar, L. Žlajpah, and A. Ude. Optimizing camera placement in agile robotic cells for visual inspection. In *Proc. RAAD 2025*, Ljubljana, Slovenia, 2025.
- [4] M. Deniša and A. Ude. Sequence optimization in multi-camera robotic visual inspection. In *Proc. IEEE/SICE International Symposium on System Integration (SII 2026)*, Nagoya, Japan, 2026. Accepted for publication.
- [5] U. M. Erdem and S. Sclaroff. Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements. *Computer Vision and Image Understanding*, 103(3):156–169, 2006.
- [6] T. Gašpar, M. Deniša, P. Radanovič, et al. Smart hardware integration with advanced robot programming technologies for efficient reconfiguration of robot workcells. *Robotics and Computer-Integrated Manufacturing*, 66:101979, 2020.
- [7] M. Held and R. M. Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10(1):196–210, 1962.
- [8] Y. Koren and M. Shpitalni. Design of reconfigurable manufacturing systems. *Journal of Manufacturing Systems*, 29(4):130–141, 2010.
- [9] I. Kovač. Cardan Joint, 4 2018.
- [10] G. Olague and R. Mohr. Optimal camera placement for accurate reconstruction. *Pattern Recognition*, 35(4):927–944, 2002.
- [11] T. Schouwenaars, E. Feron, and J. How. Integer programming as a general solution methodology for path-based optimization in robotics. *Journal of Aerospace Computing, Information, and Communication*, 2(4):89–112, 2005.