

ROBOTSKO PODPRTO PIPETIRANJE Z ROBOTSKIM SISTEMOM ANDREW+

Aleksandar Stefanov, Roman Kamnik, Janja Dermol-Černe, Aleš Belič

Izvleček:

V tem strokovnem članku je opisan razvoj aplikacije za avtomatizacijo postopka pipetiranja v farmacevtskih laboratorijih. Poudarek je na avtonomni prilagoditvi Onelab protokolov za programiranje robota Andrew+. Aplikacija vključuje uporabniku prijazen vmesnik za spreminjanje, izvajanje in spremljanje protokolov in poseben razdelek za izdelavo laboratorijskih poročil, združljivih z interno podatkovno zbirko. Temeljito testiranje v laboratorijskem okolju je pokazalo obetavne rezultate in sposobnost aplikacije, da razbremeni operaterje ponavljajočih se opravil, povezanih z robotom. Uspešno izvajanje te tehnologije pomeni pomemben napredek pri optimizaciji farmacevtskih laboratorijskih procesov, saj zagotavlja večjo učinkovitost in sprošča dragocene človeške vire za bolj kompleksne naloge.

Ključne besede:

Farmacevtska avtomatizacija, robotsko pipetiranje, farmacevtska robotizacija

1 Uvod

Postopki v farmacevtskih laboratorijih, kot so filtracija beljakovin, gensko sekvenciranje in obdelava vzorcev, vključujejo manipulacijo s številnimi tekočimi vzorci [1]. Ti postopki zahtevajo izjemno natančnost in točnost. Takšne zahteve lahko dosežemo z integracijo naprednih tehnologij, kot so robotski sistemi.

Ključni del vseh omenjenih postopkov je pipetiranje. Ročno upravljana pipeta je osnovno orodje za ravnanje s tekočino, vendar je njena učinkovitost odvisna od upravljavca, saj zahteva ročno nastavitve za vsako prilagoditev volumna tekočine in ročno pipetiranje vsakega vzorca posebej v posodice, kar je tudi časovno zamudno. Upravljanje velikega števila bioloških vzorcev s temi ročnimi napravami postane nepraktično in zamudno [2].

Izhajajoč iz tega je bilo sčasoma razvitih veliko avtonomnih pipetnih sistemov, kot so: robotski sistem Andrew+ [3], Tecan Fluent [4], Hamilton Microlab STAR Line [5] itd. Čeprav takšni robotski mehanizmi v farmacevtskih laboratorijih zelo optimizirajo procese, je njihova odvisnost od ročnega programiranja, ki ga izvaja operater, za različne naloge še vedno pomemben vidik. Operaterji morajo robot

programirati pred vsako nalogo, kar je lahko za laboratorijske raziskovalce naporno.

Glede na ta izziv je cilj naše raziskave razbremeniti in poenostaviti delovni proces operaterjev, kar jim omogoča, da se učinkoviteje osredotočijo na svoje znanstveno delo.

V raziskavi smo razvili programsko ogrodje, ki odpravlja potrebo po ročnem programiranju operaterja za vsako robotsko nalogo. Operater preprosto vnese želene parametre, nato pa naše orodje avtomatično izvede programiranje in konfiguracijo robota ter aktivno spremlja izvajanje poskusa. Po končanem poskusu orodje ustvari izčrpno poročilo, ki se nemoteno poveže z laboratorijsko podatkovno bazo za učinkovito upravljanje in analizo podatkov. Predlagano programsko orodje smo uspešno integrirali in testirali v laboratoriju podjetja Novartis.

2 Metodologije

V naši raziskavi je bil uporabljen robotski mehanizem Andrew+ [3]. Programiranje robota je bilo omogočeno s posebnimi protokoli, podprtimi v okolju Onelab [6]. Poleg tega smo izkoristili zmogljivosti tehnologije API (Application Programming Interface) in uporabili programski jezik Python za celovito optimizacijo celotnega postopka.

Strojna oprema - robot Andrew+

Pipetirni robot Andrew+ predstavlja eno izmed naj-sodobnejših rešitev na področju laboratorijske avtomatizacije.

Aleksandar Stefanov, prof. dr. Roman Kamnik, univ. dipl. inž., oba Univerza v Ljubljani, Fakulteta za elektrotehniko, Janja Dermol-Černe, Aleš Belič, oba Novartis, d. o. o., Mengeš

© The Authors 2025. CC-BY 4.0

Struktura in zgradba robota Andrew+

Robot Andrew+ ima mehansko strukturo z dvema rokama, pri čemer je ena roka kot fiksno stojalo za pipete, druga pa je premična. Premična roka ima štiri prostostne stopnje, sestavljena je iz treh rotacijskih in enega translacijskega sklepa. Robotski sistem vsebuje RGB-kamero, ki je nameščena v roki, in optični senzor, vgrajen v bazo robota. Na premični roki je nameščeno še dvoprstno prijemalo z magnetom. Komunikacija z računalnikom je omogočena preko ethernet povezave. Podrobna struktura s sestavnimi deli je prikazana na *sliki 1*.

Programiranje robota Andrew+

Programiranje robota v programski opremi Onelab omogočajo protokoli. Onelab zagotavlja vsestransko programsko okolje, ki uporabnikom omogoča ustvarjanje protokolov (programskih korakov) neposredno v programski opremi s preprosto grafično metodo »povleci in spusti« (ang. drag-and-drop). Protokoli se lahko oblikujejo tudi preko datotek CSV ali XML. Čeprav je programiranje v grafičnem okolju uporabniku prijazno za hiter začetek, lahko postane zamudno ob pogostem ustvarjanju novih protokolov. Nasprotno pa programiranje v tekstovni obliki preko datotek CSV in XML zahteva naprednejše razumevanje struktur protokolov v teh formatih. V



Slika 1: Predstavitev strojne opreme (struktura robotskega mehanizma)



Slika 2: Konfiguracija krmiljenja in programiranje robota

našem kontekstu je bilo znanje izvajanja protokolov preko CSV in XML ključno, saj se celotno programsko okolje nanaša na te formate za nemoteno delovanje. Pri tem je pomembna funkcija dinamični protokol. Ta v okolju Onelab omogoča enostavno določanje zelenega števila vzorcev za obdelavo (pipetiranje). Z avtonomnim posodabljanjem vseh korakov sistem inteligentno prilagodi protokol, ohranja osnovne značilnosti in omogoča spremembo števila vzorcev, iz katerih pipetiramo. Vendar je ta funkcionalnost v programu Onelab omejena na izbiro števila vhodnih vzorcev, brez možnosti določitve števila in količine izhodnih vzorcev pri vsakem koraku, kar je ključno za našo raziskavo. Zato smo poiskali in izvedli lastno rešitev za vključitev možnosti, kot je predstavljeno v naslednjem poglavju. Končna konfiguracija programiranja in krmiljenja robota je prikazana na *sliki 2*.

Tehnologija REST API in povezljivost s programskim okoljem Onelab

HTTP RESTful API-ji uporabljajo protokol HTTP in sledijo načelom REST (Representational State Transfer), poudarjajo način brez stanja (ang. stateless) in interakcijo na podlagi virov s standardnimi metodami HTTP (GET, POST, PUT, DELETE). Za izmenjavo podatkov se običajno uporabljata formata JSON ali XML.

V naši študiji uporabljeni vmesnik API je Onelab API, strukturiran kot HTTP RESTful API v obliki JSON, ki ponuja raznolike funkcionalnosti, vključno s spremljanjem priključenih naprav, izvajanjem in spremljanjem poskusov ter pridobivanjem informacij o vzorcih in metapodatkih [7].

V naši aplikaciji je ključni vmesnik Onelab API, ki omogoča komunikacijo med našo programsko opremo, zasnovano za avtonomno spreminjanje protokola, in robotskim sistemom. API uporabljamo za pridobivanje metapodatkov o laboratoriju, protokolih, izvorni različici protokola za spreminjanje ter nalaganje, nameščanje in izvajanje izboljšanih protokolov. Prav tako ima API pomembno vlogo pri spremljanju poskusov in pridobivanju ključnih podatkov (vzorca ID, laboratorijske opreme ID, količina vzorca itd.) za pripravo poročil, ki ustrezajo interni podatkovni zbirki družbe Novartis.

Zahteve za programsko aplikacijo

Zahteve, ki so jih opredelili raziskovalci, odgovorni za načrtovanje, in operaterji, odgovorni za njihovo izvajanje, so bile:

- ▶ nastavev poljubnega števila vhodnih in izhodnih vzorcev in nastavev poljubnega volumna;
- ▶ samostojna rekonfiguracija in programiranje gibanja robota;
- ▶ nadzor in spremljanje stanja trenutnega ekspe-

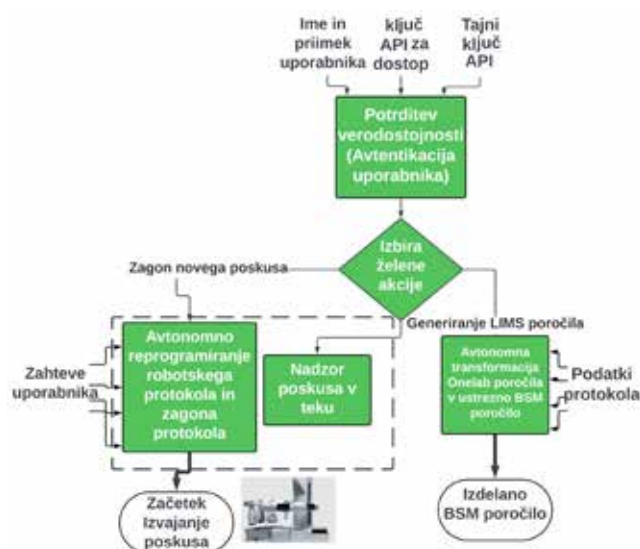
rimenta in robota in

- ▶ pridobitev poročila o eksperimentu in njegova transformacija v obliko, ustrezno za Novartisovo podatkovno zbirko.

Za izvajanje zahtevanih pogojev naše aplikacije smo uporabili vmesnik API. Razlog za to je, da programska oprema Onelab ne omogoča konfiguracije več spremenljivk (število vhodnih vzorcev, število izhodnih vzorcev, volumni, predloga poročila itd.). Ker smo želeli večjo avtonomijo pri konfiguraciji parametrov, je bila edina možnost integracija aplikacije preko vmesnika API.

Programska oprema

Aplikacijo sestavljajo štirje glavni segmenti, od katerih vsak opravlja različne funkcije: avtentikacija uporabnika, avtonomno reprogramiranje, namestitve in zagon protokola, nadzor tekočega poskusa ter ustvarjanje poročil LIMS (Laboratory information management system). V naslednjih podpoglavjih podrobneje obravnavamo vsakega od teh štirih glavnih segmentov. Diagram poteka celotne aplikacije je predstavljen na *sliki 3*.



Slika 3 : Glavni diagram poteka aplikacije

Avtentikacija uporabnika

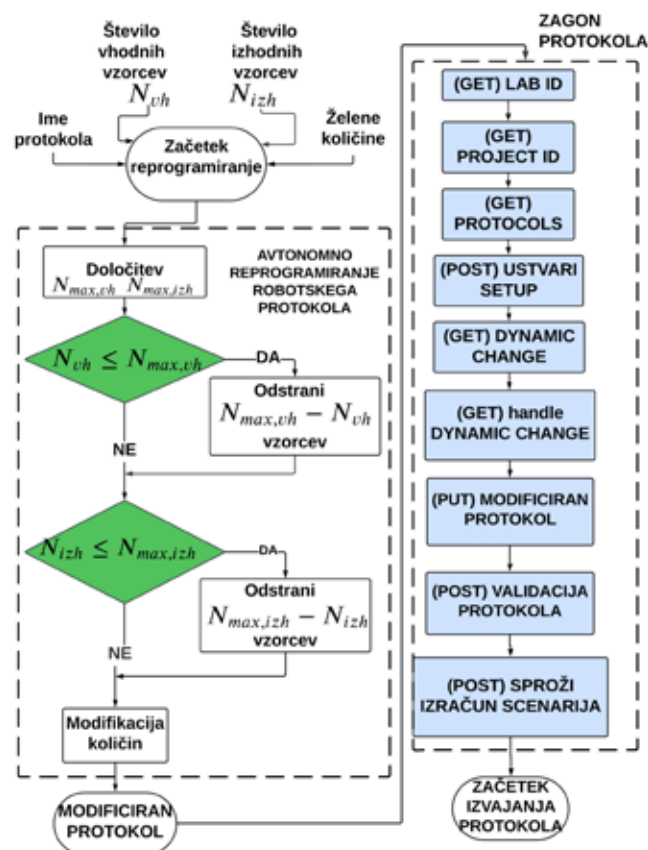
Postopek avtentikacije je ključni začetni korak v delovnem procesu aplikacije. Služi kot obvezno varovalo, ki preprečuje nepooblaščen dostop do aplikacije. Postopek avtentikacije zahteva vnos posebnih poverilnic uporabnika, vključno z njegovim imenom, priimkom, ključem za dostop do ključa API in tajnega ključa API. Ti ključi, ki so edinstveni in zasebni za vsakega uporabnika v Onelab laboratoriju, imajo ključno vlogo pri varovanju in preverjanju pristnosti zahtevkov za API ter zagotavljajo zanesljiv in varen mehanizem za preverjanje pristnosti uporabnikov.

Avtonomno reprogramiranje, namestitve in zagon protokola

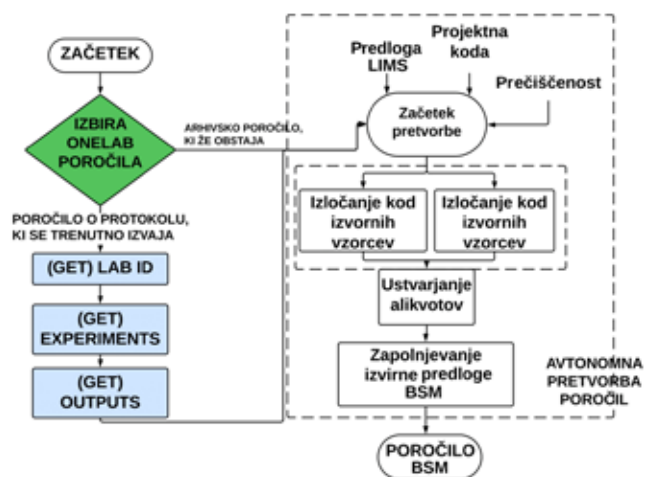
Ta faza temelji na manipulaciji protokolov, zapisanih v obliki CSV in njihovem prenosu na robot s pomočjo vmesnika API. Sestavljena je iz dveh glavnih stopenj: modifikacije in zagona poskusa. V fazi spreminjanja, ko uporabnik vnese svoje pogoje za poskus, programska oprema identificira dele protokola, ki jih je treba spremeniti, in na podlagi vnesenih pogojev prilagodi obstoječi protokol. Ko je spremenjeni protokol pripravljen, se izvede veriga ukazov API s ciljem, da se nastavitev robota potrdi, namesti in izvede spremenjeni protokol. Podrobnejši potek programskih korakov je predstavljen na *sliki 4*.

Avtonomno generiranje poročil LIMS

V tem segmentu aplikacije se izvede pretvorba poročila iz Onelab formata v poročilo LIMS. Onelab poročilo za tekoči poskus je mogoče pridobiti preko vmesnika API ali pa ga lahko uvozi uporabnik. Nato algoritem izvede vrsto ukazov za mani-



Slika 4 : Diagram poteka avtonomnega reprogramiranja, namestitve in zagona protokola. Zeleni in prozorni bloki spreminjajo protokola. Modri bloki v diagramu predstavljajo ukaze API, ki se izmenjujejo med razvito aplikacijo in strežnikom Onelab, med zagonom. Spremenljivka N_{vh} predstavlja število vhodnih vzorcev, $N_{vh,max}$ največje število vhodnih vzorcev v poskusu, N_{izh} število izhodnih vzorcev in $N_{izh,max}$ največje število izhodnih vzorcev.



Slika 5 : Diagram poteka avtonomne pretvorbe iz Onelab poročila v poročilo LIMS. Zeleni in prozorni bloki izvedejo pretvorbo. Modri bloki v diagramu predstavljajo ukaze API, ki se izmenjujejo med našo aplikacijo in strežnikom Onelab pred začetkom pretvorbe.

pulacijo podatkov, ki podatke poskusa prilagodijo ustrezni obliki LIMS. Podroben potek programa je prikazan na *sliki 5*.

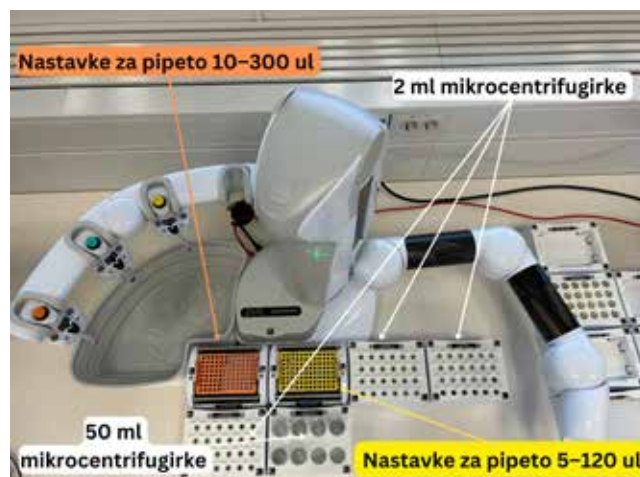
Testiranje

Testiranje in ovrednotenje našega programskega okolja sta potekala v Novartisovem laboratoriju, kjer smo izvedli obsežno testiranje dveh posebnih protokolov: pipetiranje iz 50 ml centrifugirk v 2 ml mikrocentrifugirke in pipetiranje iz 15 ml centrifugirk v 2 ml mikrocentrifugirke.

V obeh primerih smo poskuse konfigurirali tako, da so pri maksimalnem scenariju vključevali pipetiranje iz petih vzorcev, pri čemer iz vsakega od petih vzorcev v 12 različnih mikrocentrifugirk. Posledično so končni rezultati vsakega poskusa v maksimalnih pogojih obsegali pet skupin, od katerih je vsaka vsebovala 12 vzorcev, pridobljenih iz ustreznih izvornih vzorcev (ang. parent samples).

Nato je bila programska oprema temeljito preizkušena v štirih različnih scenarijih:

- ▶ spreminjanje samo števila vhodnih vzorcev ob ohranjanju konstantnega števila izhodnih vzorcev;
- ▶ spreminjanje samo števila izhodnih vzorcev ob ohranjanju konstantnega števila vhodnih vzorcev;
- ▶ spreminjanje tako števila vhodnih kot tudi izhodnih vzorcev in
- ▶ število vhodnih in izhodnih vzorcev ostane nespremenjeno. Ti testni scenariji so bili ključni za oceno robustnosti in prilagodljivosti razvitega programskega ogrodja v različnih eksperimentalnih pogojih. Testno okolje je prikazano na *sliki 6*.



Slika 6 : Med testiranjem uporabljena oprema

3 Rezultati

Rezultat raziskave je v celoti razvita programska aplikacija, ki jo je mogoče upravljati in spremljati preko našega grafičnega vmesnika. Grafični vmesnik je prikazan na *slikah 7* in *8*.

4 Razprava

Na podlagi testov, ki smo jih izvedli z operaterji v laboratoriju, je aplikacija pokazala obetavne rezultate, ki lahko operaterja zelo razbremenijo ponavljajočega se dela pri oblikovanju protokola za programiranje robotov. Pomemben vidik je tudi združljivost poročila LIMS z interno podatkovno bazo družbe Novartis. Pred našo aplikacijo so morali operaterji po vsakem poskusu ali ročno vnesti vse izhodne rezultate v sistem ali pa ročno spremeniti tabelo. Z uporabo naše aplikacije pa operaterji



Slika 7 : Grafični vmesnik aplikacije. Slika zgoraj – avtentikacija uporabnika, slika spodaj – generiranje poročila LIMS.



Slika 8 : Grafični vmesnik aplikacije. Slika zgoraj – reprogramiranje in zagon poskusa na robotu, slika spodaj – spremljanje stanja poskusa v teku.

dobijo v celoti generirano poročilo, ki ga lahko neposredno vstavijo v podatkovno zbirko. Aplikacija je bila razvita v tesnem sodelovanju z operaterji, da bi bila njena uporaba čim bolj preprosta. Grafični uporabniški vmesnik je preprost in intuitiven za uporabo, operaterju pa ni treba podrobno poznati posebnosti programov Onelab in Andrew+.

5 Zaključek

Razvita aplikacija je praktična rešitev za avtomatizacijo postopka pipetiranja v farmacevtskih laboratorijih. Aplikacija ne le poenostavlja spreminjanje protokolov in programiranje robotov, temveč tudi ustvarja poročila LIMS, ki so združljiva z notranjimi podatkovnimi zbirkami. Uspešno laboratorijsko testiranje poudarja njeno učinkovitost in zagotavlja

obetavno pot za zmanjšanje bremena ponavljajočih se opravil za operaterje in povečanje splošne operativne učinkovitosti.

Literatura

- [1] Kong F. in ostali: Automatic liquid handling for life science: a critical review of the current state of the art, *J Lab Autom.*, 2012.
- [2] L. P. Bheemavarapu in ostali: Intelligent Pipetting System Towards Automatic Liquid Handling Applications, *IEEE International Symposium on Medical Measurements and Applications (MeMeA)*.
- [3] Andrew+ The Pipetting Robot, Andrew Alliance. Dostopno na: <https://www.andrewalliance.com/pipetting-robot/> (Dostopano: 25. 11. 2024).
- [4] Tecan Fluent, Tecan. Dostopno na: <https://lifesciences.tecan.com/fluent-laboratory-automation-workstation> (Dostopano: 25. 11. 2024).
- [5] Hamilton Microlab Star liquid handling system, Hamilton STAR. Dostopno na: <https://www.hamiltoncompany.com/automated-liquid-handling/platforms/microlab-star> (Dostopano: 25. 11. 2024).
- [6] Design and Execute Laboratory Protocols: OneLab, Andrew Alliance. Dostopno na: <https://www.andrewalliance.com/laboratory-software/> (Dostopano: 25. 11. 2024).
- [7] Onelab API documentation. Onelab, Andrew Alliance. Dostopno na: <https://hub.andrewalliance.com/enterprise/docs/onelab-api-1.19.0.pdf> (Dostopano: 25. 11. 2024).

Robotic pipetting with the Andrew+ robotic system

Abstract:

This technical paper details the development of an application aimed at automating the pipetting process in pharmaceutical laboratories. The focus is on autonomous adapting of Onelab protocols for programming the Andrew+ robot. The application incorporates a user-friendly interface for protocol modification, execution and monitoring, as well as a dedicated section for generating laboratory reports compatible with the internal database. Thorough testing within the laboratory setting demonstrated promising results, showcasing the application's ability to alleviate operators from repetitive tasks associated with the robot. The successful implementation of this technology marks a significant advancement in streamlining pharmaceutical laboratory processes, offering enhanced efficiency and freeing up valuable human resources for more complex tasks.

Keywords:

Pharmaceutical automation, robotic pipetting, pharmaceutical robotization