

# Mobil robotkar felhasználása a mezőgazdaságban

## Mobile robotic manipulator for precision agriculture

KÖLLŐ Magor Örs<sup>1</sup>, MOLNÁR Szilárd<sup>2</sup>, Dr. TAMÁS Levente<sup>3</sup>

<sup>1</sup>Kolozsvári Műszaki Egyetem, Kolozsvár, Memorandumului utca 28, 400114  
, www.utcluj.ro, kollo.va.magor@student.utcluj.ro

<sup>2</sup>Kolozsvári Műszaki Egyetem, molnar.fe.szilard@student.utcluj.ro

<sup>3</sup>Kolozsvári Műszaki Egyetem, Levente.Tamas@aut.utcluj.ro

### Abstract

*In this paper a mobile manipulator arm for automated harvesting is proposed. This device can be useful in such places, where the human resources are too expensive. The main parts of the assembly are a four-wheeled robot and a robotic arm, depth sensing camera and embedded GPU device. The device can navigate through obstacles without hitting them, and find a path to the targeted plant. With a camera it can recognize the fruits, and based on visual servoing it can crop them.*

### Kivonat

*Ebben a dolgozatban bemutatunk egy mobil robot kart, amely nagy segítséget nyújthat az agrikultúrában, főleg olyan helyeken, ahol az emberi munkaerőhiány van. Az eszköz főbb alkotórészei egy négykerekű mozgó robot, valamint egy robotkar. A szerkezet képes elnavigálni az akadályok között és amikor talál egy növényt, képes felismerni rajta a gyümölcsöt, majd azt leszedni, és elszállítani.*

**Kulcsszavak:** robotkar, mezőgazdaság, képfelismerés, mesterséges intelligencia, navigáció

## 1. BEVEZETÉS

A növények termesztése mindig is létszükséglete volt az embernek. Ennek ellenére, még mindig sokat fejlődhetünk ezen a téren, sőt további problémák is sorra előjönnek. Az egyik ilyen nehézség, hogy az eddigi vidéki lakosság nagy része, aki eddig többnyire önellátó módon termesztette a növényeit, felhagy ezzel a tevékenységgel, és több hasznot hajtó foglalkozást folytat. Ez pedig azt eredményezte, hogy egyre kevesebb egyén kezében van a gazdaság, akik egyrészt a munkaerőhiány miatt, másrészt a nagyobb profit érdekében, silányabb minőségű terményeket kínálnak eladásra, nagy mennyiségben [1]. Ezen a téren nagy előrelépést jelent, az időközben megjelenő automatizációs rendszerek bevezetése [2].

Egy másik probléma, az olyan területek megművelése, ahol kifejezetten nehéz a növénytermesztés, például a sivatagos, illetve minimálisan csapadékos helyeken. Ezekben a területeken nemcsak a körülmények nehezítik tevékenységet, de az emberek sem rendelkeznek megfelelő képzéssel.

Elsősorban ilyen helyzetekre keres megoldást a az AGRO AI [3] nevezetű projekt, amelynek célja, hogy különböző gépi tanulási technikákat alkalmazva, nagy mennyiségben jó minőségű élelmiszert állítsanak elő. Ezen projekten belül 8 különböző terményt tanulmányoznak, mint például paradicsom, burgonya, rizs vagy bab. Vizsgálják a megfelelő talaj összetételét, a terményt támadó kártevőket, betegségeket. És nem utolsósorban a lehetséges vásárlókat is előre tájékoztatják a valószínű hozamról, amely elejét veszi a eldobandó árunak vagy a kiürült üzleteknek, optimalizálva a termelési láncot.

Az AGRO AI nyújtotta példa adta az ötletet, hogy tanulmányképpen készítsünk egy prototípust egy mobil robotkarról, amelyet majd továbbfejlesztve felhasználhatunk olyan körülmények között, ahol nehéz lenne emberi munkásokat alkalmazni. A robotunk képes önmagától navigálni, anélkül, hogy akadályoknak ütközne, illetve a kamerája segítségével felismeri a növényeket, amelyeket a kar fogó mechanikája leszed illetve egy dobozba rak.

A dolgozatban bemutatjuk, hogyan működik a szerkezet. Továbbá ismertetjük a különálló elemekről, amelyekből felépül az egész, a felhasznált szoftverről és az algoritmusokról is.

## 2. FELHASZNÁLT HARDWARE

### 2.1 Mozgató robot: Pioneer 3-AT

A dolgozatunkban leírt mobil robotkar több különböző eszközből épül fel: Az egész alapját egy 4 kerekű robot adja, a PIONEER 3-AT [4] nevű robot pontosabban.

Ezen a roboton található mind a négy kerék saját elektromotorral rendelkezik, ezért a kerekeke teljesen függetlenül forognak. Ezek a motorok elég erősek ahhoz, hogy legalább 5 kilogrammot könnyedén elszállítson. Ez fontos szempont mivel több eszközt is rászerteltünk. Továbbá az alkalmazott gumiabroncsok típusa illetve a felfüggesztés magassága lehetővé teszi a navigálást akár egyenetlen terepen is. Robot rendelkezik továbbá odometrikus szenzorral, amely szerepe a visszajelzés arról, hogyan mozdul el a robot, vagyis melyik tengelyeken észlelhető gyorsulás. Ezekből az adatokból nem csak sebességet lehet állítani, de ki lehet számolni a robot pozícióját a kiindulási ponthoz viszonyítva. A robot látható az 1. ábrán.



1. ábra  
*PIONEER 3-AT mobil robot*

A robot mellett említést teszünk továbbá a robotra felszerelt LiDArról is, amelyet a végső szerkezetben nem használtunk, de rajta maradt a roboton, erre szereltük rá a robotikus kart, illetve több kezdeti tesztben is segítségünkre volt. A LiDAR modellje Sick LMS200.

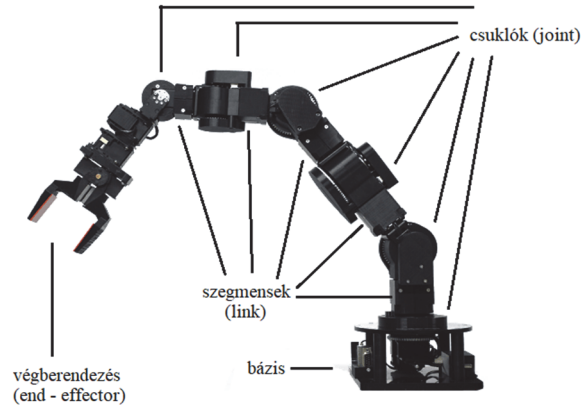
### 2.2 Kamera: Pico Zense DCAM710

A robot célja, hogy az őszi betakarításkor a térben látható gyümölcsöket (alma, szilva, körte stb) képes legyen felismerni és összegyűjteni. Ehhez szükség van egy kamerára, amely segítségével felismerjük az összegyűjtésre szánt zöldeket, gyümölcsöket. Fontos szempont ugyanakkor, hogy ezeket a tárgyakat ne csak felismerjük, hanem a térben képesek legyünk elhelyezni is, ezért egy 3D kamerát használtunk, a Pico Zense DCAM710-et. Ez egy Time-of-Flight (ToF) [5] érzékelést használó kamera, amely a hagyományos 2D RGB kép mellett mélységképet is felvesz, mivel kivetített fénysugarak segítségével megméri a távolságot az érzékelt környezet és a kamera között. Az érzékelő rendszer fényjeleket bocsájt ki, amelyek rövid időn belül visszaverődnek az érzékelendő tárgyakról és visszatérnek a szenzor vevőegységéhez. A 3D kamera előnye, hogy a hagyományos kép - és tárgyfelismerési technikákat is használhatjuk, de az érzékelt tárgyakhoz 3D koordinátákat és távolsági információkat is szolgáltat.

### 2.3 Robotkar: Cyton Gamma 1500

A robotkar egy olyan irányított rendszer, amely előre megadott pályán képes mozogni, a pálya mentén feladatokat képes elvégezni és tárgyak manipulálására, mozgására alkalmas eszközként szolgál. A robotkarok több részből épülnek fel: a robotkar alapja a bázis, amely az a pont, ahol a rendszer el van helyezve a térben. Emellett van a végberendezés (end-effector) amely a céltárgy mozgatásáért, manipulálásáért felelős eszköz, ez teremt kapcsolatot a robot és környezete között. Több végberendezés létezik, mi egy megfogót (gripper) használtunk. A végberendezést és a bázist csuklók és szegmensek kötik össze. A szegmensek merev testek amelyeket a csuklók mozgatnak és így a csuklók mozgása irányítja a robotkart is.

A dolgozatban leírt robot esetében egy Cyton Gamma 1500 [6] típusú robotkart használtunk, amely összesen 7 szabadsági fokkal rendelkezik, így egyszerűbb és komplex mozgások elvégzésére egyaránt alkalmas. A robotkar látható a 2. ábrán. A Robai által forgalmazott robotkar tömege 3 kg, összesen 1.2 kg mozgatására alkalmas (egyszerűbb mozgások esetében 1.5 kg-t is képes mozgatni). A robotkar összesen 76 cm hosszú, és 68 cm-es hatótávolsággal rendelkezik. A standard végberendezés, gripper 3.5 cm nyitási tartománnyal bír.



2. ábra  
*Cython Gamma 1500-as robotkar*

#### 2.4 Vezérlő egység: Jetson Nano

A tárgyfelismeréshez egy neurális hálózatra van szükség, amely mélységi tanulás segítségével (deep learning) felismer meghatározott tárgyakat. A tanítást nagy teljesítményű számítógépeken végzik, viszont ha rendelkezésre áll egy hosszasan betanított hálózat már nincs szükség olyan magas számítási teljesítményre, hogy ezeket a hálózatokat implementáljuk. Ennek köszönhetően a hálózatokat alacsony fogyasztású, akkumulátorral működtetett beágyazott eszközökön is lehet használni, jelentősen növelve a kisebb robotok mobilitását, feladatainak komplexitását. Mi erre a célra az Nvidia által fejlesztett és forgalmazott Jetson Nano-t használtuk, amely a beépített GPU-nak köszönhetően alkalmassá vált a valós idejű tárgyfelismerő algoritmusok futtatására, ugyanakkor alacsony fogyasztással rendelkezik és kis helyet foglal el, ami fontos szempont egy mobil robot esetében.

Az összeszerelt robotot a 3. ábrán láthatjuk.

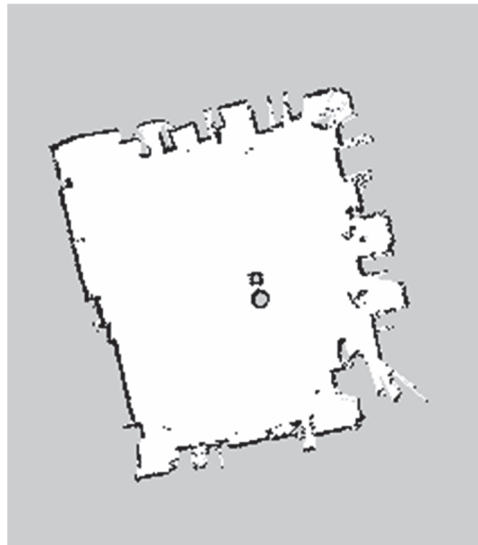


3. ábra  
*Az összeszerelt robot*

### 3. VEZÉRLÉS

#### 3.1 Navigáció

A navigáció a robot egyik kiemelendő karakterisztikája. Ennek a modulnak az agyát a a Robotic Operating System (ROS) egyik könyvtára alkotja, a Navigation stack [7]. Ez elsősorban lidarra támaszkodik, ugyanis a navigáció legelső lépése a feltérképezés, azaz SLAM Gmapping (Simultaneous Localization and Mapping) algoritmus [8] [9]. Ebben a elindítjuk a robotot és a LiDAR-t, vagy egy mélység kamerát, amely képéből kinyerünk egy LiDAR-ra jellemző pont felhőt. Amennyiben megfelelően adjuk meg az algoritmusnak a kamera elhelyezkedését a robothoz viszonyítva, távirányítást használva végigjárjuk a feltérképezni kívánt helyiséget a robottal. Ahogy a szenzor gyűjtögeti az információkat, úgy az algoritmus automatikusan egymásra helyezi az adatokat, és elkészíti a helyiség térképét. A mi helyiségünk szkennelt térképe látható a 4. ábrán. A térkép valamelyest könnyebben érthető lesz ha megvizsgáljuk az 5. ábrát.



4. ábra  
*A terem térképe*



5. ábra  
*A terem lemodellezett képe*

A kapott térképet illetve a szenzorokból érkező információt felhasználva, a robot képes elnavigálni saját magát úgy, hogy az egyetlen külső parancs a célpont meghatározása.

Ehhez az algoritmusnak szüksége van egy jelenlegi helyzetre, amit az AMCL modul határoz meg. Továbbá különböző paraméterekre, például a robot alapterületére (ami körülbelül egy 40x40

centiméteres négyzet, amennyiben a kar nincs valamilyen irányban kinyújtva), emellett pedig egy előre meghatározott minimális távolságra az akadályoktól. Az algoritmus legenerál ezekből egy úgynevezett költség térképet, ahol az akadályok nagy, a környezetük pedig közepes költséget jelent. A szabad út kis költséggel jár. Az algoritmus útvonal tervezője pedig kiszámolja a legkisebb költséggel járó útvonalat, ami aztán bejárásra kerül.

### 3.2 Képfelismerés

A tárgyfelismerésre az SSD Mobilenet V2 [10] [11] neurális hálózatot használjuk, a kamera a 2 dimenziós RGB képeket továbbítja a neurális hálózatnak, amely architektúrája szerint egy single – shot – detector (magyarázat erről) és a képeket elemezve bejelöli a felismert tárgyakat és egy behatároló négyzetet rajzolunk a tárgy köré. A tárgyfelismerést a futási idő optimalizálása érdekében az Nvidia által fejlesztett TensorRT-n futtatjuk. Ezeket az adatokat visszaküldjük a kamerának amelynek a ToF érzékelőjét használva kapjuk meg a céltárgy elhelyezkedését a térben és ezeket a koordinátákat továbbítjuk a robotkarnak.

Fontos szereppel bír a kamera és a robotkar közötti kalibráció is, amely segítségével a kamera által észlelt koordinátákat transzformáljuk a robotkar koordináta rendszerébe, ezáltal a robotkar és a mozgását vezérlő MoveIt! [12] platform tudja a céltárgy elhelyezkedését a robotkar bázisához képest, így tudunk egy trajektóriát számolni és azt végrehajtani. A kar - kamera kalibráció egy összetett probléma, amely kulcsfontosságú a szerkezet pontosságát tekintve. A kamera elhelyezésére két megközelítés is van: amikor a kamerát a térben helyezik el (eye to hand) és amikor a kamerát a végberendezés közelében, a karra helyezik (eye in hand [13]). Mi az utóbbi megoldást választottuk és terveztünk, majd 3D nyomtató segítségével kinyomtattunk egy kamera tartót, amit a karra szereltünk fel. Ezt követően a transzformáció változóit Konstantinos Danidiilidis [14] kettős kvaterniók algoritmusát felhasználva számoltuk egy ROS bővitmény segítségével [15].

### 3.3 A kar mozgása

A kar mozgatására a MoveIt! mozgástervező platformot használjuk, amely könnyen összekapcsolható az ROS rendszerrel, megkönnyítve a kommunikációt a komponensek között. A kar modelljét, meglévő URDF (Universal Robot Description Format) valamint SRDF (Semantic Robot Description Format) formátumokat kiegészítve tettük használatra alkalmassá a MoveIt!-ban, ezeken keresztül a mozgástervező platformban biztonsági korlátokat állíthatunk be a karnak, így megelőzve a szegmensek és csuklók közötti ütközéseket. A mozgások megtervezése a Cartesian motion planner [16] segítségével történik, az algoritmus egy trajektóriát generál a kezdeti és a végső pont között. Ahogy a robotkar megkapta a céltárgy koordinátáit, ezt a statikus transzformáció segítségével a saját koordináta rendszerére vetíti le, megtervezi és végrehajtja a mozgást. Minden alkalommal egy kezdeti, inicializált pozíciót vesz fel, ebből figyeli meg a környezetében a tárgyakat. A mozgás eredménye, hogy a végberendezés a megadott koordinátákat éri el a bázis koordináta rendszerében. Mielőtt elérné a cél tárgyat, a megfogó (gripper) a kamera által észlelt szélességben nyílik ki, a kar ezt követően éri el a céltárgyat és fogja meg majd egy előre megadott pozícióban rakja le az elszállításhoz.

### 3.4 Modulok közötti kommunikáció

A modulok többsége úgynevezett ROS topikokon keresztül kommunikál egymással. Ezek a topikok bizonyos időközönként ki vannak küldve egy “közös térbe”, ahonnan akármelyik modul lekérheti a tartalmukat, ami egy bizonyos típusú üzenet. Az üzenet típusa küldőként változik, attól függően, hogy mi az üzenet tartalma. Például a kamera egy képet küld ki üzenetként, de a robot irányításához egy másik, specifikus parancsra jellemző üzenet architektúrárt kér mint üzenet. Ez az üzenet tartalmazza a sebességeket, valamint a kért elmozdulást mindegyik tengelyen.

## 4. A MÓDSZER TESZTELÉSE

A robotot egyelőre nem próbáltuk ki éles helyzetben, mivel ez nem egy végső termék, sokkal inkább prototípus. a teszt helyszíne egy terem volt, ami elég nagy ahhoz, hogy kellően berendezhessük akadályokkal, de hely is marad, ahol a robot navigálni tud. És mivel ebben a teremben nem folyik növénytermesztés, mesterségesen kellett termő növényt előállítani, ami szobanövényre aggatott

gyümölcsökből állt, például alma vagy banán. A gyümölcszedési procedúrának két fázisa van. Ezeket külön tárgyaljuk.

Az első fázis a navigáció. Ahogy már korábban is leírtuk, először feltérképezzük a termet, úgy hogy végigvezetjük a robotot rajta, amivel elkészül egy térkép. Utána a robot képes navigálni a teremben. Egyelőre a kamera nem ismeri fel a teljes növényt, csak a gyümölcsöt, ezért a térképen bejelöltük a gyümölcs helyét. Ezután a robot elindul és körülbelül 10-20 centis pontossággal meghatározza a saját helyzetét a tárgyakhoz képest. Ez első körben jó eredmény, de a továbbiakban mindenképpen finomításra szorul, mivel ha több ideig működik a robot, és több célponthoz kell eljusszon, idővel ezek a hibák összeadódnak, még akkor is ha az érzékelőkből érkező jel segít a pontosság tartásához, főleg ha azt vesszük figyelembe, hogy az akadályok helyzete változhat, illetve a szabadban a viszonyítási pontok is megbízhatatlanabbak.

Ha a robot megérkezett a növényhez, kezdetét veszi a második fázis. A tárgyfelismerés megbízhatóan működik, a használt neurális háló közelebbi felvételek esetén ~90% -os pontossággal, távolabbi képeken ~79% - os pontossággal működik, 640 x 480 -as képfelbontás mellett. Ezt követően a ToF érzékelővel határozzuk meg a kamera és a tárgy közötti távolságot, amelynek a középpontját néhány centiméteres pontossággal visszatéríti, és ellenőrizzük hogy a céltárgy a kar hatótávolságán belül helyezkedik-e el. A tárgy pontos méreteit egyelőre szintén a 2D kamera által meghatározott koordináták behatároló négyzetek számoljuk ki, ez a módszer könnyen adhat pontatlan eredményeket. Ha a tárgy méretei megvannak, a megfelelő szélességre nyitjuk ki a megfogót (grippert) és a végberendezést a céltárgyhoz mozgatjuk. A helyes kalibráción múlik, hogy a kar képes-e elérni a céltárgyat, valamint a mozgás komplexitásán. A tesztek során több problémával is szembesültünk, az első az volt, hogy a vezérlésre használt Jetson Nano nem volt képes egyszerre a tárgyfelismerő és a trajektória tervező algoritmusok futtatására, ezért először meghatározzuk a céltárgy koordinátáit, leállítjuk a tárgyfelismerő algoritmusokat és ezt követően tervezzük meg a kar trajektóriáját, amely hosszas várakozási időt vesz igénybe. Másik probléma az egyszerű mozgások túlkomplikálása volt, valamint a helytelen kalibrációból kifolyó pontatlanságok. A céltárgy formája sem mindegy, ahogyan a keménysége sem (egy szilvát könnyedén kiprésel a megfogó). Az esetek többségében a kar eléri a céltárgyat de a mérési és számítási hibák miatt az almát vagy más céltárgyat az esetek többségében már nem fogja meg. A későbbiekben szeretnénk a megfogót helyettesítő, alternatív megoldásokat is kipróbálni, mint egy speciális gyümölcs gyűjtő zsák, amit az iparban is használnak. Emellett a pont felhőből kapott adatokat szeretnénk használni, egy neurális hálózat segítségével a céltárgy lemérésére a 2D kamera helyett.

## 5. VÉGSŐ KIÉRTÉKELÉS

Ebben a dolgozatban bemutatunk egy mobil robotikus kar prototípusát, amelyet fel lehet használni gyümölcsök zöldségek betakarításánál. Az általunk létrehozott egyszerű környezetben tesztelve megfelelő eredményeket értünk el, viszont egy további fejlesztésekre van szükség. Első sorban a navigációt kell sokkal pontosabbá tenni, itt különböző más mesterséges intelligenciára alapuló módszereket is alkalmazni lehet, illetve a LiDAR kép helyett a teljes mélységképet is lehetne tanulmányozni, nem is beszélve egy speciális beltéri GPS modulról. A kar és a kamera között egy sokkal pontosabb kalibrációra, valamint több adatra van szükség, emellett a jövőben a céltárgyak méreteit a 3D kamera által létrehozott pont felhőben szeretnénk meghatározni, hogy minél pontosabb információval rendelkezünk a 100%-os betakarítás céljából.

## 6. KÖNYVÉSZET

- [1] L. F. P. Oliveira, A. P. Moreira and M. F. Silva, "Advances in Agriculture Robotics: A State-of-the-Art Review and Challenges Ahead," *Robotics*, vol. 10, no. 2, 2021.
- [2] S. Bachche, "Deliberation on Design Strategies of Automatic Harvesting Systems: A Survey," *Robotics*, vol. 4, no. 2, pp. 194-222, 2015.
- [3] "AGRO AI," [Online]. Available: <https://www.f6s.com/agroai1>. [Accessed 13 09 2021].
- [4] A. M. Robots, "PIONEER3AT datasheet," 2011. [Online]. Available: <https://www.generationrobots.com/media/Pioneer3AT-P3AT-RevA-datasheet.pdf>. [Accessed 13 09 2021].
- [5] "3D Time of Flight (ToF)," [Online]. Available: <https://www.analog.com/en/applications/technology/3d-time-of-flight.html>. [Accessed 13 09 2021].

- [6] "Cyton Gamma 1500 Arm Specifications," [Online]. Available: <https://www.cdiweb.com/datasheets/ro-bai/cyton-gamma-1500-arm-specifications-2015.pdf>. [Accessed 13 09 2021].
- [7] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey and K. Konolige, "The Office Marathon: Robust Navigation in an Indoor Office Environment," in *International Conference on Robotics and Automation*, 2010.
- [8] G. Grisetti, C. Stachniss and W. Burgard, "Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters," *IEEE Transactions on Robotics*, vol. 23, pp. 34-46, 2007.
- [9] G. Grisetti, C. Stachniss and W. Burgard, "Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling," *In Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005.
- [10] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017.
- [11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *Computer Vision -- ECCV 2016*, Springer International Publishing, 2016, pp. 21-37.
- [12] D. Coleman, I. A. Şucan, S. Chitta and N. Correll, "Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study," *Journal of Software Engineering for Robotics*, vol. 5, no. 1, p. 3–16, 2014.
- [13] G. Flandin, F. Chaumette and E. Marchand, "Eye-in-hand / eye-to-hand cooperation for visual," *IEEE Int. Conf. on Robotics and Automation, ICRA '00.*, pp. 41-2746. , 2000.
- [14] K. Daniilidis, "Hand-Eye Calibration Using Dual Quaternions," *The International Journal of Robotics Research*, vol. 18, no. 3, pp. 286-298, 1999.
- [15] "Hand-Eye Calibration," [Online]. Available: [https://ros-planning.github.io/moveit\\_tutorials/doc/hand\\_eye\\_calibration/hand\\_eye\\_calibration\\_tutorial.html](https://ros-planning.github.io/moveit_tutorials/doc/hand_eye_calibration/hand_eye_calibration_tutorial.html). [Accessed 13 09 2021].
- [16] V. Munoz, I. Garcia-Morales, J. Morales, J. Gomez-DeGabriel, J. Fernandez-Lozano and A. Garcia-Cerezo, "Adaptive cartesian motion control approach for a surgical robotic cameraman," *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 3, pp. 3069-3074, 2004.