

Robot pentru industria farmaceutică

*Manipularea și umplerea
recipientelor farmaceutice*

Elevă: Szabo Ilinca
Clasa: a XI-a

Secțiunea Roboți

Robot pentru industria farmaceutică

-Manipularea și umplerea recipientelor farmaceutice

Cuprins

I. Utilitate practică.....	4
II. Mecanică.....	6
II.1.Complexitate.....	10
SISTEMUL DE MIŞCARE.....	10
SISTEMUL DE PRINDERE.....	12
SISTEMUL DE UMPLERE.....	13
III.Electronica.....	14
Microcontrollere.....	14
Componete electronice active.....	14
Senzori.....	18
IV. Software.....	19
Principiile de mișcare.....	19
BIBLIOGRAFIE.....	26

I. Utilitate practică

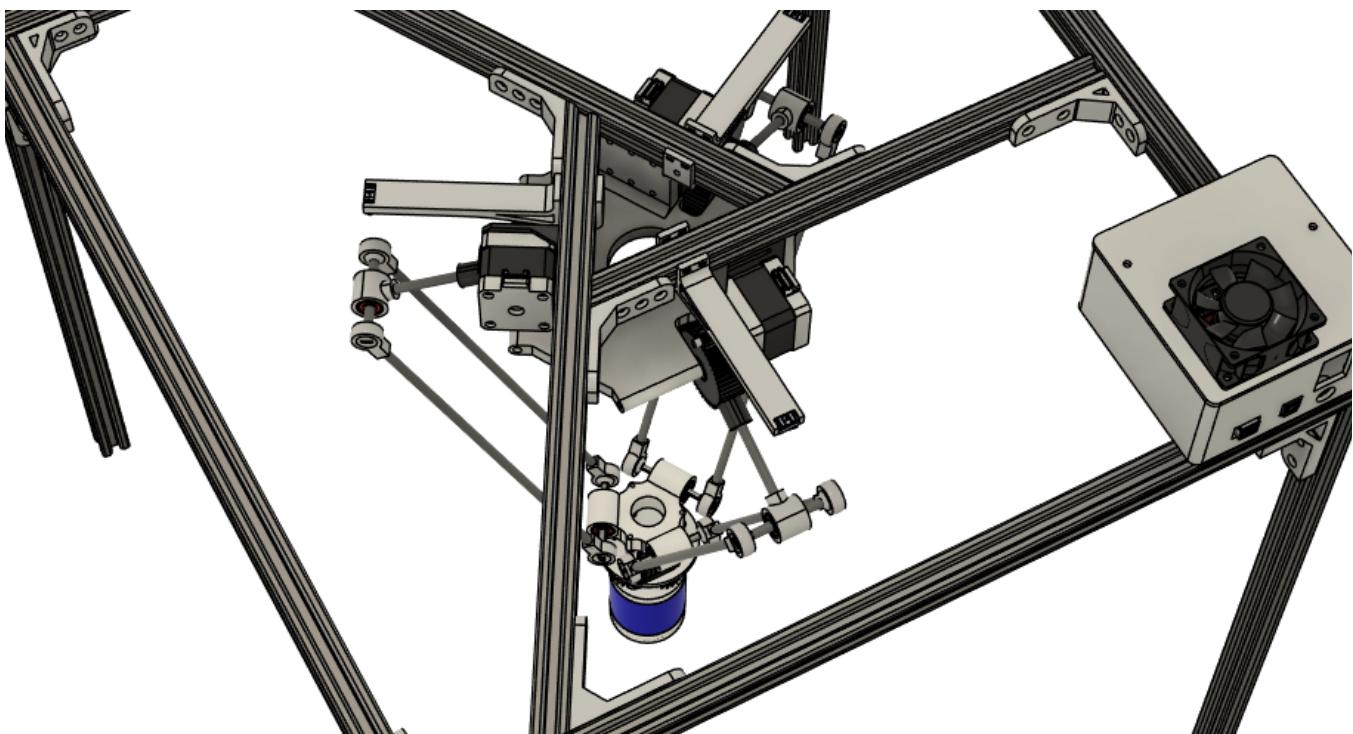
Sistemul propus este un robot paralel de tip delta, proiectat pentru automatizarea procesului de manipulare a recipientelor de tip sticlă, dozare de medicamente solide și aşezare organizată a acestora pe un suport de colectare. Scopul său principal este de a integra într-o singură unitate funcțională operații care, în fluxurile clasice de producție farmaceutică, sunt realizate de mai multe echipamente distințe. Astfel, robotul reduce complexitatea liniei de producție, ocupă un spațiu fizic mai mic și permite o operare mai eficientă, cu o nevoie redusă de intervenție umană.

Prin utilizarea unei arhitecturi delta, se obține un sistem cinematic rapid, cu mișcări precise și curzive în plan tridimensional, ideal pentru sarcini de manipulare de mare viteză. Cinematica paralelă oferă un raport excelent între rigiditate și viteză, iar masa redusă a elementelor mobile permite accelerări mari fără pierderi de precizie. Acest aspect este esențial în aplicațiile farmaceutice, unde timpii de ciclu scurți și toleranțele stricte sunt cerințe de bază.

Robotul este proiectat pentru operare în medii controlate și are capacitatea de a executa secvențe de lucru într-un spațiu preconfigurat. Poziționarea sa relativă în raport cu alte echipamente este configurația manuală, iar ciclurile de operare sunt programate pe baza unor coordonate fixe. Acest mod de operare este suficient pentru scenarii de producție repetitive, unde poziția recipientelor și a dozatorului nu se modifică frecvent. În plus, arhitectura software și hardware permite extinderea ulterioară prin integrarea unor module de viziune sau senzori de feedback, fără modificări structurale majore.

Capul terminal al robotului este modular, ceea ce permite înlocuirea rapidă în funcție de tipul și dimensiunea recipientelor utilizate. Acest lucru este important pentru flexibilitatea liniei de producție, în special în fabrici care procesează mai multe tipuri de medicamente sau ambalaje. Dozatorul este și el ajustabil, atât din punct de vedere mecanic, cât și al parametrilor de operare, pentru a putea administra diverse forme, dimensiuni și densități ale medicamentelor solide — cum ar fi comprimatele, capsulele sau tabletele efervescente.

Sistemul este proiectat pentru funcționare continuă (24/7), cu o structură mecanică fiabilă, folosind componente standardizate, ușor de întreținut sau înlocuit. Aceasta îl face potrivit pentru integrarea în linii industriale automate, unde disponibilitatea echipamentului este esențială pentru menținerea ritmului de producție.





Utilitatea practică a sistemului se extinde și dincolo de aplicațiile farmaceutice. Datorită modularității și principiilor generale de funcționare, acesta poate fi adaptat pentru linii de ambalare a suplimentelor alimentare, produse cosmetice sau chiar elemente de micro-electronică, unde manipularea precisă a obiectelor de mici dimensiuni este o cerință frecventă. Pentru spațiile reduse, cum ar fi farmacii sau cabinete medicale, sistemul poate fi reconfigurat într-o variantă compactă, mobilă, care să funcționeze autonom sau semi-autonom, în funcție de specificul aplicației.

Pe termen scurt, robotul poate contribui la creșterea consistenței procesului de ambalare, la reducerea variabilității în dozare și la eliminarea erorilor rezultate din manipularea manuală. În același timp, permite operatorilor umani să fie relocați spre activități de supraveghere și control, ceea ce reduce oboseala operațională și risurile asociate cu sarcinile repetitive.

Prin caracteristicile sale constructive și funcționale, robotul se încadrează în tendințele actuale din industria 4.0, în care flexibilitatea, modularitatea și autonomia echipamentelor devin criterii esențiale pentru optimizarea producției și reducerea pierderilor. Fiind dezvoltat cu un nivel moderat de autonomie și un cadru deschis pentru extindere, sistemul reprezintă o soluție viabilă atât pentru aplicații industriale, cât și pentru scenarii de cercetare sau dezvoltare de prototipuri în domeniul tehnologiilor medicale.

II. Mecanică

Robotul este o elaborare a unui sistem de tip Delta (paralel), proiectat pentru aplicații care necesită viteză, precizie și repetabilitate ridicată, fiind ilustrat în schema geometrică (Fig. 2.1.). Acesta este alcătuit din trei brațe identice ($B_i W_i X_i$, unde i din mulțimea $\{1, 2, 3\}$), fiecare conectat la platforma fixă (baza, O) și convergând spre o platformă mobilă de dimensiuni reduse (Y), numită efector final. Configurația paralelă a acestui robot implică faptul că toate brațele lucrează simultan pentru a poziționa efectorul în spațiul de lucru, spre deosebire de roboții seriali unde fiecare axă este dependentă de cele anterioare.

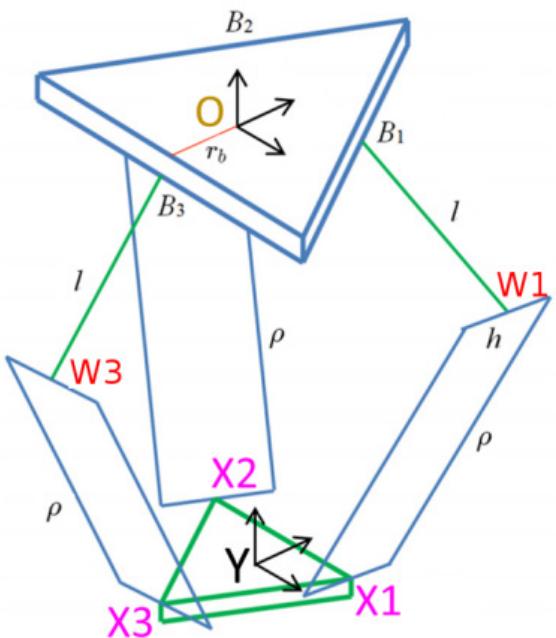


Fig. 2.1. - geometria 3D a robotului

Elementul distinctiv al acestui tip de robot este utilizarea unor mecanisme tip paralelogram articulat, care limitează mișcarea platformei mobile doar la translații liniare în planul tridimensional (X , Y și Z , conform notațiilor consacrate carteziene), eliminând complet rotațiile sau inclinările acesteia. Astfel, efectorul final se deplasează precis într-un volum de lucru bine definit, fără să-și modifice orientarea.

Baza robotului este poziționată deasupra zonei active, ceea ce permite montarea tuturor motoarelor de acționare pe cadrul fix. Această caracteristică are un impact major asupra performanței: brațele robotului pot fi realizate din materiale ușoare (ex. fibră de carbon sau aluminiu), reducând masa totală a componentelor în mișcare. În consecință, robotul poate atinge foarte mari și poate executa mișcări cu accelerări semnificative, fiind ideal pentru operațiuni de tip pick-and-place, ambalare, sortare sau asamblare rapidă.

Structura cu trei brațe articulate conectate simultan la platforma mobilă asigură o rigiditate crescută a sistemului, ceea ce îmbunătățește semnificativ precizia poziționării și repetabilitatea mișcărilor. Totuși, această construcție impune anumite limite asupra dimensiunii volumului de lucru, care este relativ compact comparativ cu alte arhitecturi robotice, cum ar fi cele seriale cu șase grade de libertate.

Avantaje:

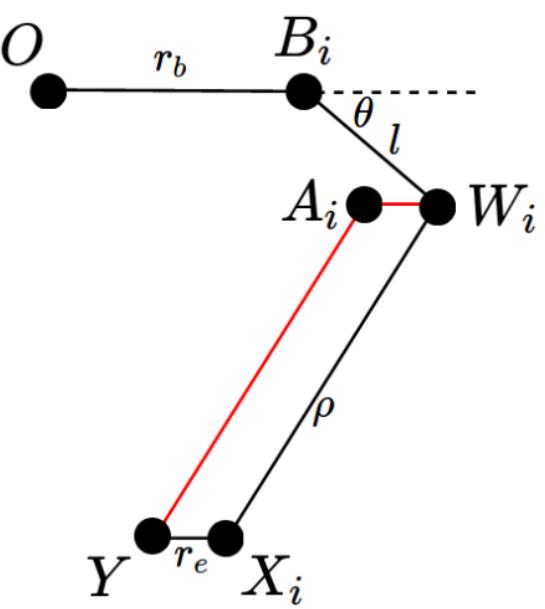
Viteză și accelerări ridicate – Datorită masei reduse a componentelor mobile, robotul poate executa mișcări extrem de rapide, fiind ideal pentru procese repetitive de mare viteză.

Precizie și repetabilitate excelente – Mișcarea sincronizată a celor trei brațe și rigiditatea structurală asigură o precizie constantă în poziționarea efectorului final.

Eficiență energetică – Fiindcă motoarele sunt montate pe partea fixă, efortul mecanic este redus, ceea ce scade consumul de energie și uzura componentelor.

Design compact – Structura simplificată și volumul de lucru bine definit îl fac potrivit pentru linii de producție automatizate cu spațiu limitat.

Flexibilitate în aplicații – Robotul poate fi echipat cu diverse unelte (grippere, ventuze, senzori etc.) pentru a se adapta ușor la diferite procese industriale.



Determinarea zonei accesibile

Notătii și valori:

- R - Raza bazei (de la îmbinare la centru) - 75mm
- r - Raza platformei efectoare (de la îmbinare la centru) - 30mm
- L_1 - Lungimea brațului superior - 100mm
- L_2 - Lungimea brațului inferior - 300mm

(Aceste valori au fost stabilite experimental și prin efectuarea calculelor următoare)

Presupunem că centrul bazei se află la coordonatele (0,0,0) (origine) într-un reper cartezian XYZ cu originea în punctul O. Vom aproxima zona de lucru a robotului la o formă geometrică tridimensională cunoscută, și anume o bază de con (frustum), care se extinde în jos, spre -Z, de sub bază.

- **Raza superioară** - Cea mai mică suprafață accesibilă pe XY
- **Raza inferioară** - Cea mai mare suprafață accesibilă pe XY
- **Înălțimea** - Lungimea mișcării verticale maxime posibile

Raza superioară

$$R_{\text{sup}} = \max (0, (R - r) - (L_1 + L_2)) = \max (0, 45 - 400) = 0\text{mm}$$

Din moment ce raza nu poate fi negativă, o vom considera 0, adică brațele se pliază perfect într-un punct apropiat de bază.

Raza inferioară

$$R_{\text{inf}} = (R - r) + L_1 + L_2 = (75 - 30) + 100 + 300 = 445\text{mm}$$

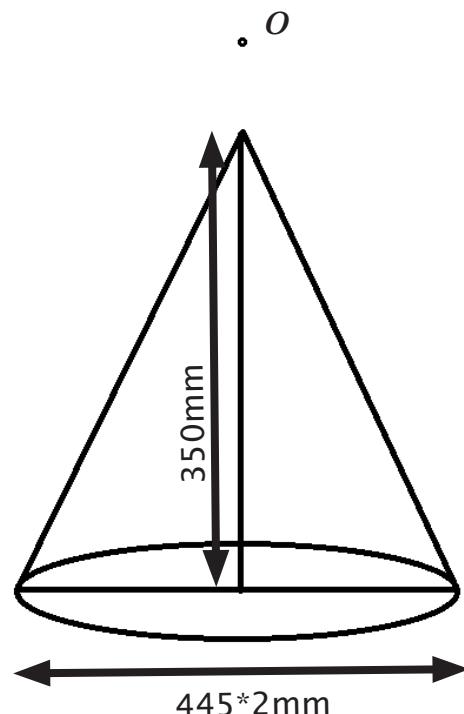
Înălțimea

$$h = Z_{\text{min}} - Z_{\text{max}} = (-400) - (-50) = 350\text{mm}$$

Volum

$$\begin{aligned}V &= 0,33 \pi h * (R_{\text{sup}}^2 + R_{\text{sup}} * R_{\text{inf}} + R_{\text{inf}}^2) \\V &= 0,33 \pi * 350 * (0^2 + 0 * 445 + 445^2) \\V &= 0,33 \pi * 350 * 198,025 \approx 72.436.583 \text{ mm}^3 \\&\approx 0,072 \text{ m}^3\end{aligned}$$

În concluzie, zona accesibilă a robotului poate fi aproximată cu ajutorul unui trunchi de con, care se dovedește a fi un caz particular de con, situat vertical la 50mm de centrul bazei spre -Z, cu o bază de 445mm și o înălțime de 350mm. Așadar, robotul are la dispoziție 0,072 m³ pentru a efectua sarcinile date, forma fiind una ideală pentru manipulare la nivelul bazei și mici ridicări.



Determinarea vitezei, puterii si preciziei

Specificațiile de bază ale motoarelor (Nema 17):

- *Cuplu de menținere* - 0.56 N·m
- *Pași/revoluție* - 200 steps/rev (1.8 deg/step)
- *Max RPM* - fără încărcătură - aprox 3000 RPM
- *Max RPM real* - 1000 RPM
- *Microstepping*

Pentru a menține precizia la unghiurile limită am ales să folosesc un raport de 5:1.

Cuplu ieșire: Cuplu de menținere * 5 = 0,56 * 5 = 2,8 N*m

Viteză ieșire: Max RPM real / 5 = 1000 / 5 = 200 RPM

Viteza platformei efectoare

Calculul vitezei unghiulare:

$$\omega = \Delta \vartheta / \Delta t = 2\pi * \text{RPM} / 60 = 2\pi * 200 / 60 \approx 20.94 \text{ rad/s (s}^{-1}\text{)}$$

Calculăm viteza tangențială a capătului:

$$v = L_1 * \omega = 0,1 \text{ (m)} * 20,94 \text{ (s}^{-1}\text{)} = 2,094 \text{ m/s}$$

Totuși, având în vedere inerția și greutatea sistemelor atașate, putem approxima la 0,6-1,2 m/s.

Puterea platformei efectoare

Forța per motor:

$$F = \text{Cuplul de ieșire} / L_1 = 2,8 / 0,1 = 28 \text{ N}$$

Așadar, forța totală va fi în intervalul 40-50N luând în considerare alinierea optimă a motoarelor.

Precizie

Pași compleți (fără microstepping): aprox. 0.63mm

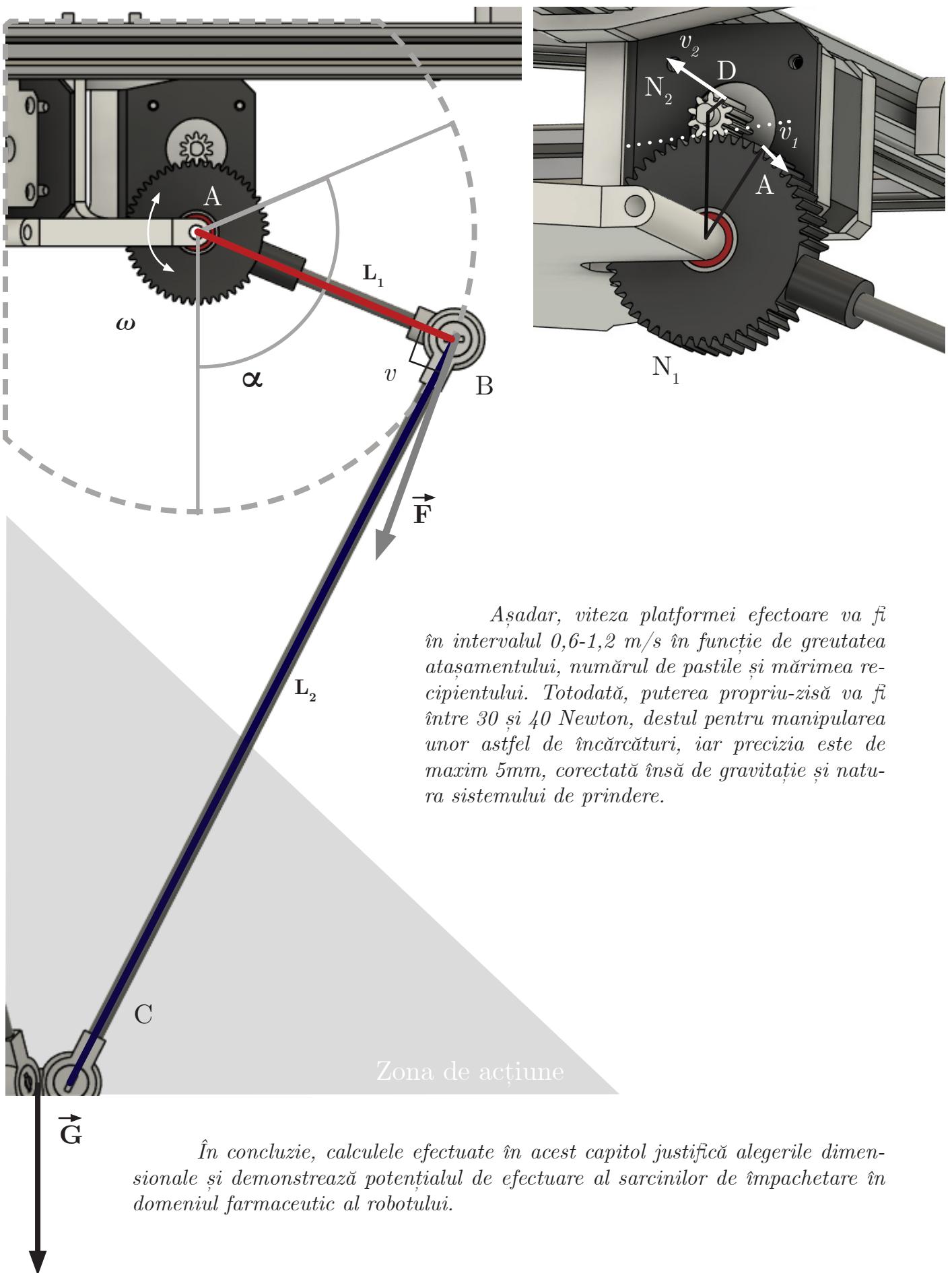
1/16 fără microstepping: aprox. 0.04mm

Totuși, având în vedere toleranțele rulmenților, avem $\mathcal{E} \approx 5 \text{ mm}$, corectat însă de forța gravitațională.

Schema fortelelor - Legendă

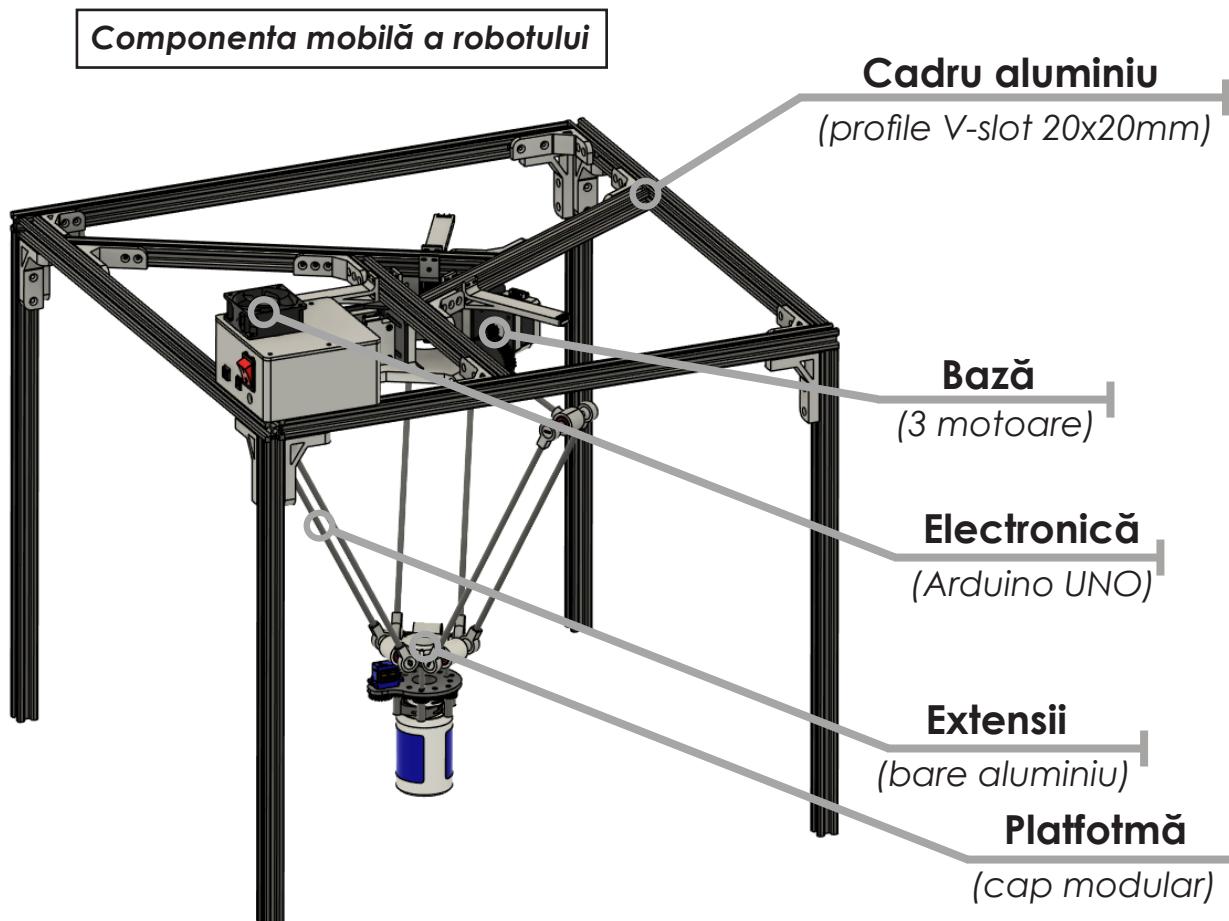
- | | |
|--|---|
| <i>A</i> - Centrul pinionului cu 50 dinți | • <i>F</i> - Cuplul de ieșire |
| <i>B</i> - Încheietura dintre brațe | • <i>G</i> - Greutatea platformei efectoare |
| <i>C</i> - Îmbinarea brațului cu platforma efectoare | • <i>ω</i> - viteza unghiulară în A |
| <i>D</i> - Centrul pinionului antrenat direct | • <i>v</i> - viteza tangențială în B |
| <i>L₁</i> - Brațul superior | • <i>N₁</i> - Numărul de dinți al pinionului D |
| <i>L₂</i> - Brațul inferior | • <i>N₂</i> - Numărul de dinți al pinionului A |
| <i>α</i> - Unghiul de acțiune al brațului superior | • |
| <i>v₁</i> - Viteza pinionului acționat direct | • |
| <i>v₂</i> - Viteza pinionului intermediu | • |
| | • |
| | • |
| | • |

Schema forțelor



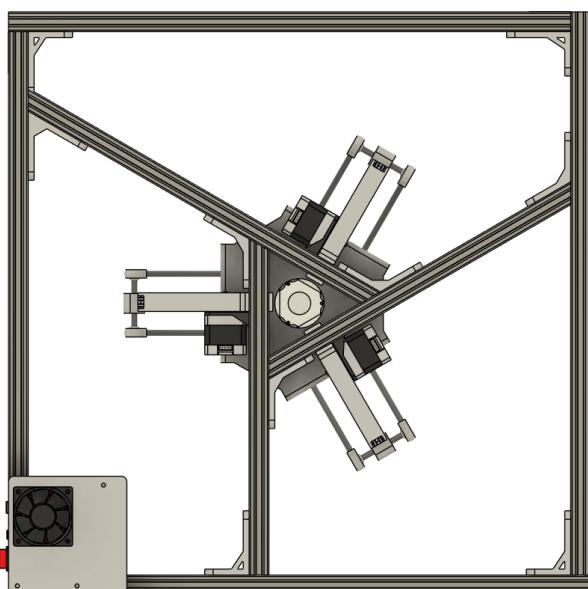
II.1. Complexitate

Robotul are în alcătuire 4 motoare cu pas (steppere) și un servo-motor micro, care pun în mișcare o varietate de sisteme adiacente:

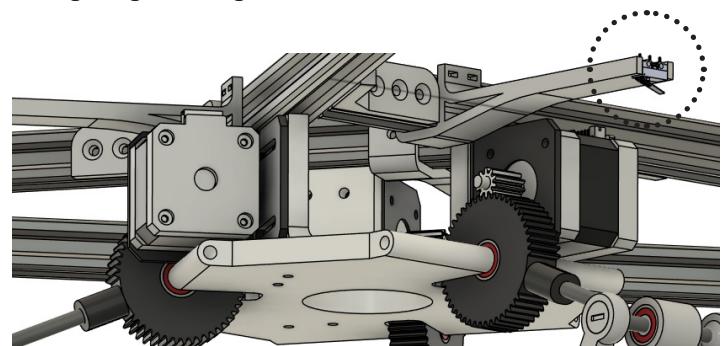


Din punct de vedere al alcătuirii componentei mobile, robotul cuprinde patru componente principale, controlate de microcontrolerul principal, organizat în zona pentru electronică:

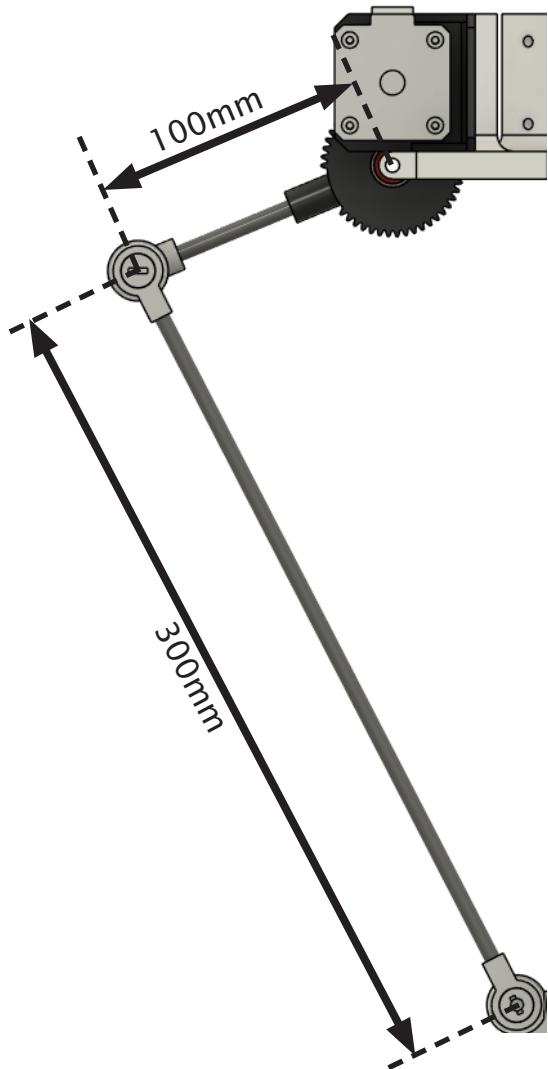
- Cadrul de aluminiu - este conceput din profile V-Slot 20x20mm. Suportul principal este construit din 4 bucăți de 570mm, iar picioarele din 4 bucăți de 490mm. Îmbinările sunt proiectate ca un sistem culisant blocat prin șurube și sunt de patru tipuri, 120, 90, 60 și 30 de grade.



- Baza - este construia pentru a controla întreaga componentă mobilă a robotului prin intermediul a trei motoare cu pas, atașate prin montarea unor suporti din aluminiu modularesi, compatibili cu stepperele Nema 17. Transmisia este de 5:1, și anume 10 la 50 de dinți, realizată prin pinioane printate 3D din PET-G.



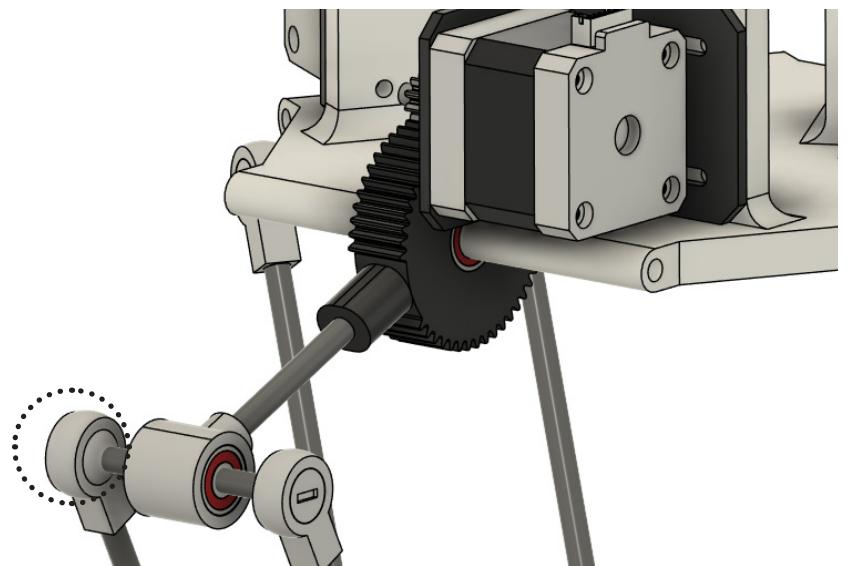
Această alegere facilitează transformarea vitezei motorului în forță pentru a menține precizia inclusiv la unghurile limită. Configurația motoarelor este în formă de triunghi echilateral pentru a garanta o mișcare uniformă și ușor de determinat matematic. În același timp, baza este prevăzută cu trei brațe dotate cu limit switch-uri (senzori de apăsare) care au rolul de a inițializa robotul într-o poziție cunoscută.



- Extensiile - sunt alcătuite din două zone:

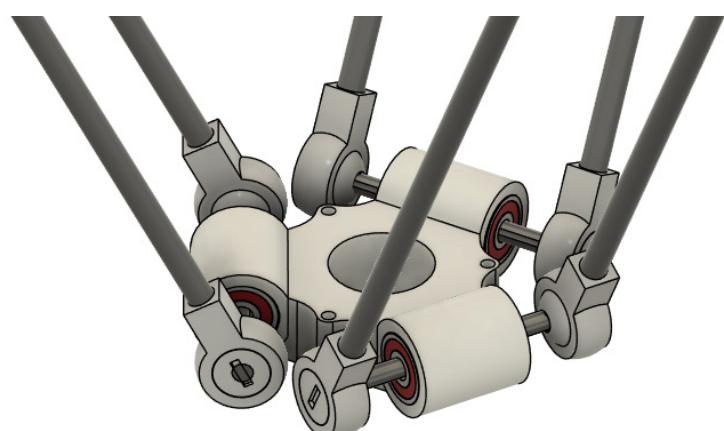
SEGMENTUL 1, cel superior, atașat la pinionul de 50 de dinți, lungime 100mm și SEGMENTUL 2, cel inferior, care conectează segmentul 1 de platformă, lungime 300mm. Lungimile brațelor de diametru 6mm dictează zona de acțiune, viteza și precizia extensiei mobile.

Legătura, atât dintre cele două segmente, cât și dintre platformă și segmentul 2, se face printr-o bară intermediară filetată M5, perpendicular amplasată pe planul determinat de cele două segmente. Legăturile dintre segmentul 2 și barele intermediare se face prin îmbinări de tip bilă, care permit mișcarea de rotație liberă a porțiunii inferioare față de cea superioară.

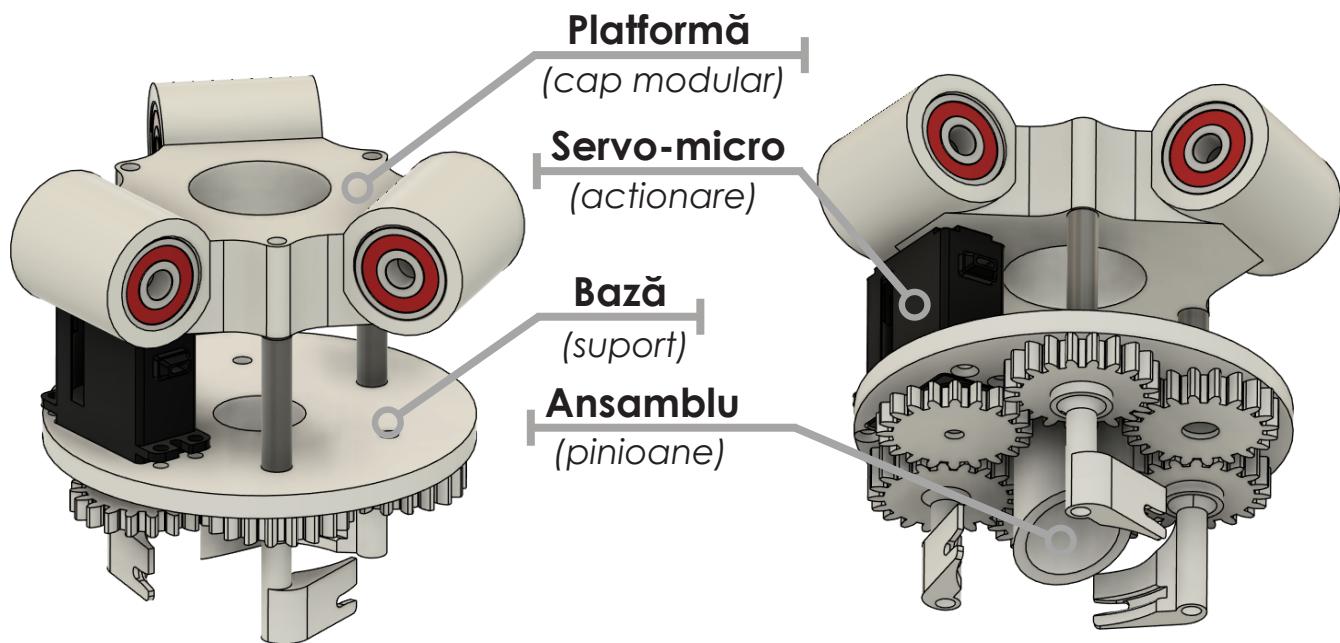


Alinierea segmentelor se face printr-o piesă în formă de T, prevăzută cu rulmenți pentru a permite rotația, în timp ce menține poziționarea relativă a brațelor.

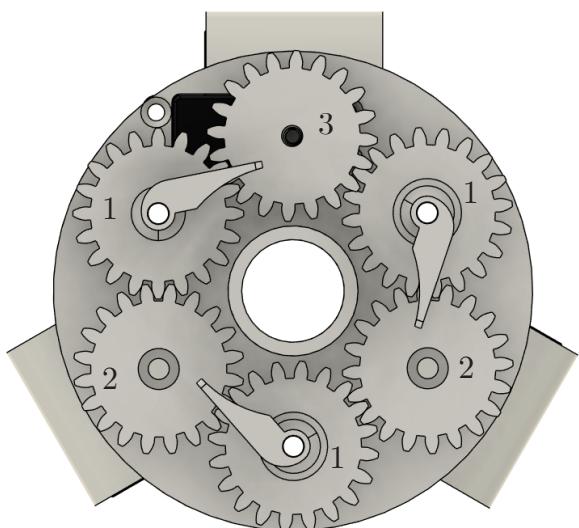
- Platforma - este prevăzută cu trei prinderi pentru cele trei extensii. Poziționarea acestei componente reprezintă rezultanta poziționărilor motoarelor anterior menționate, din acest motiv este prevăzută cu capete modulare pentru a interschimba atașamentele în funcție de tipul recipientului care urmează să fie umplut. Între această componentă și bază există un tub prin care pastilele vor face trecerea de la dozator la recipient.



Sistemul de prindere



Prinderea recipientului se face prin intermediul unui ansamblu de 6 pinioane de 3 tipuri diferite. Acesta este atașat de platforma efectoare pentru a putea fi schimbat cu ușurință în funcție de tipul recipientului. Punerea în mișcare a sistemului se face prin intermediul unui servo-motor micro.



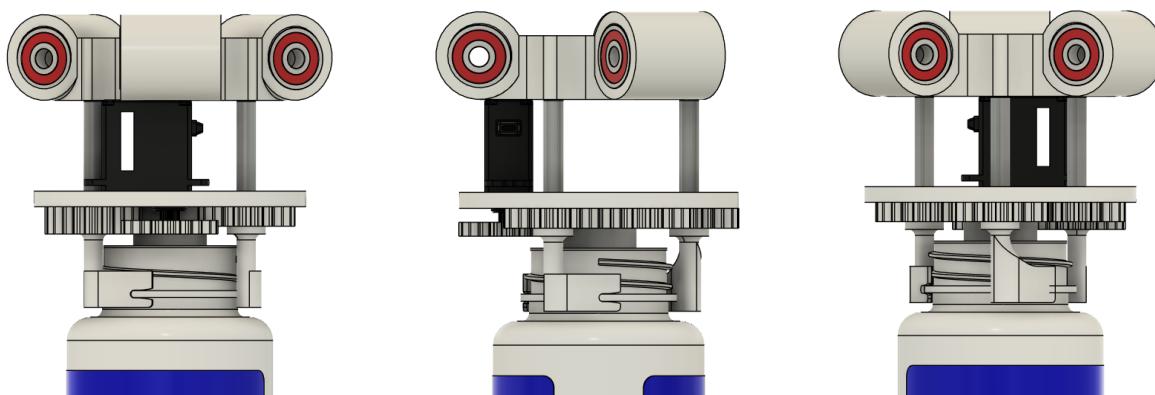
În figura alăturată este ilustrat ansamblul cu pinioane. Există 3 tipuri de pinioane:

- **Tipul 1** - este dotat cu palete pentru prinderea recipientului, matisate în cauciuc, pentru o deplasare sigură. Acestea sunt amplasate în formă de triunghi echilateral și se deplasează simultan;

- **Tipul 2** - reprezintă cele două pinioane intermedii, care asigură mișcarea roților de tip 1 în același sens. Acestea sunt așezate pe două dintre laturile triunghiului;

- **Tipul 3** - face legătura între motor și pinioanele de tip 1 și 2, acesta fiind atașat direct de servo-motor.

Baza este prevăzută cu un tub secționat diagonal la capăt, care încadrează sticla în zona potrivită înainte de preluare, minimizând posibilele erori.



Determinarea vitezei de prindere

Viteza servoului:

$$\omega = 60\text{deg} / 0.62\text{s} = 1.047\text{rad} / 0.62\text{s} = 16,88 \text{ rad/s}$$

Având în vedere că mototul funcționează la 7.4V.

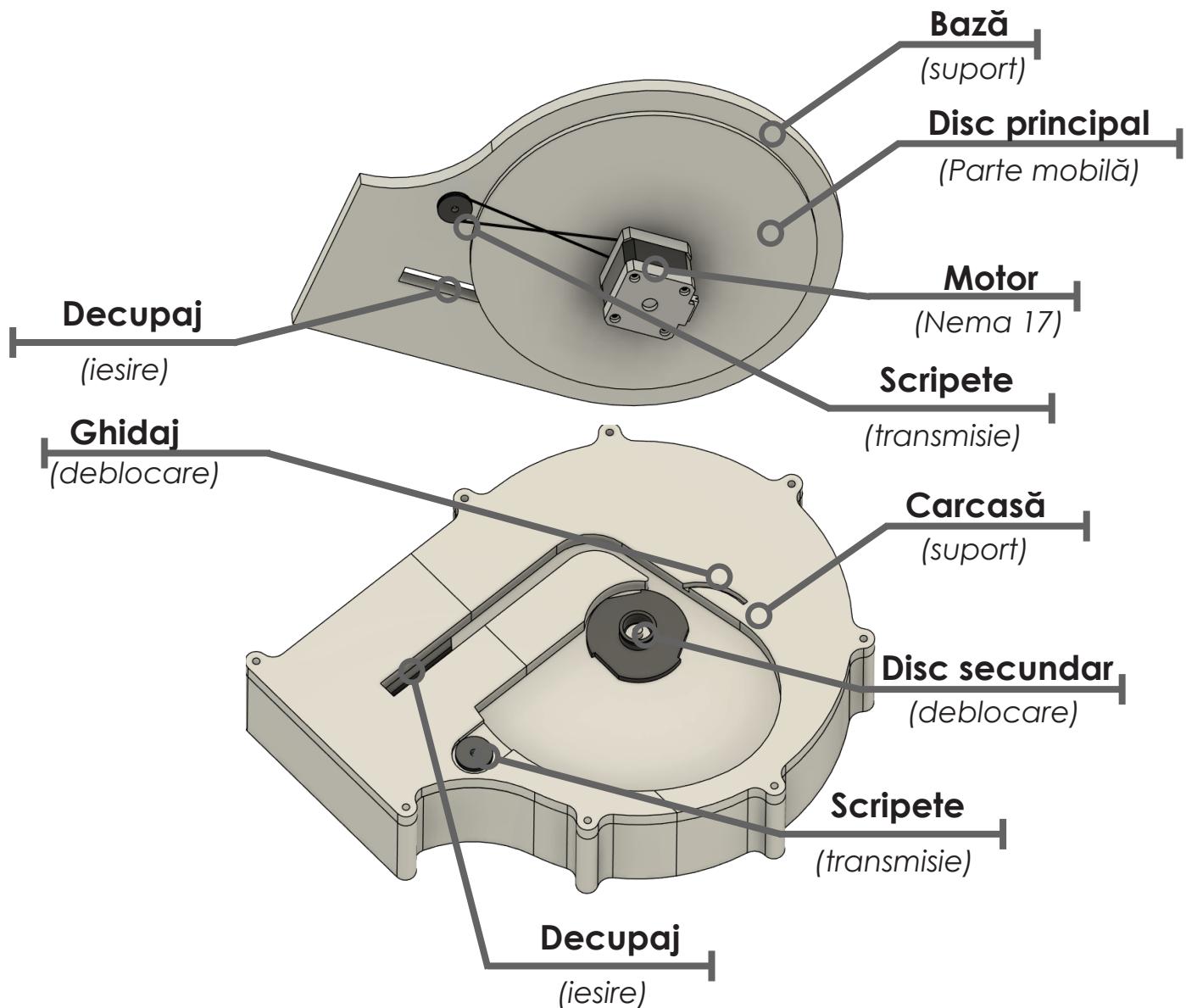
Timpul necesar deplasării pe porțiunea vizată (90deg):

$$90\text{deg} = 1,5780 \text{ rad} = \Delta\vartheta$$

$$\omega = \Delta\vartheta / \Delta t \Leftrightarrow \Delta t = \Delta\vartheta / \omega$$

$$\Delta t = 1,578 / 16,88 = 0,09 \text{ s}$$

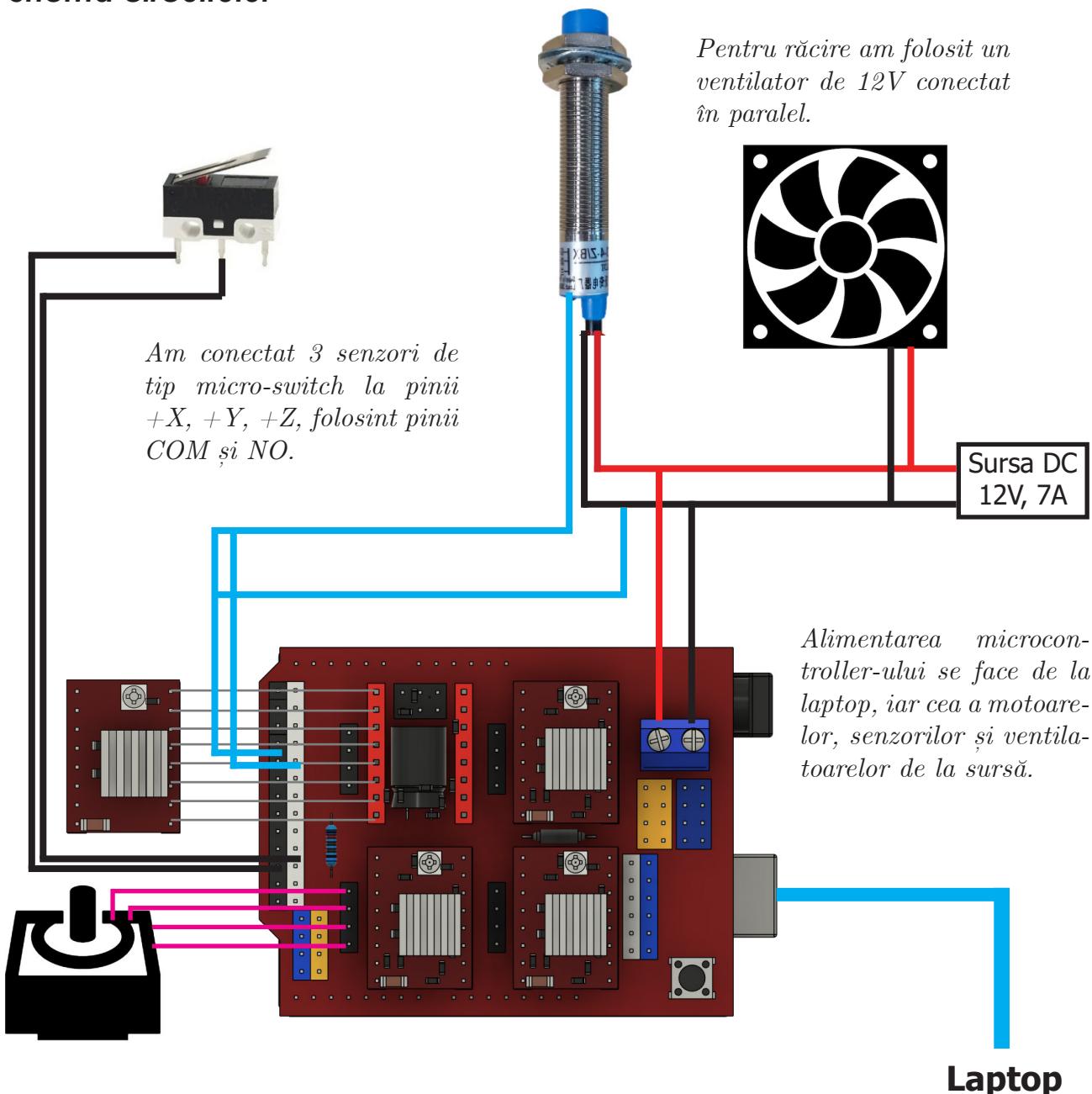
Sistemul de dozare



Dozatorul are rolul de a distribui numărul exact de capsule pentru a umple recipientul. Inițial, corpurile sunt vărsate pe discul principal, începând să fie aruncate spre canalul de ieșire, fiind aliniate de ghidaj. Pentru a preveni blocarea la zona de ieșire, platforma secundară se mișcă în sens opus pentru a arunca în spate capsulele predispuse la blocarea sistemului. Transmisia se face prin curele subțiri, ambele discuri fiind antrenate de același motor. Pornirea și oprirea sistemului se face la comanda unui senzor de distanță de înaltă precizie, care procesează numărul de pastile.

III. Electronică

Schema circuitului



III.1. Microcontrolere & componente electrice active

Din punct de vedere al componentelor electrice, la baza robotului se află o placă de dezvoltare compatibilă cu Arduino UNO R3, echipată cu un microcontroler ATmega328P (Fig. 3.1.1). Alegerea acestei plăci a fost motivată de fiabilitatea și popularitatea sa în proiectele de tip DIY și robotică educațională, fiind una dintre cele mai bine documentate și ușor de utilizat soluții pentru începători și avansați deosebitiv. De asemenea, comunitatea vastă din jurul ecosistemului Arduino oferă suport extins și numeroase exemple de aplicații, ceea ce ușurează considerabil dezvoltarea și depanarea proiectului.

Alimentarea plăcii se realizează prin intermediul unui port USB conectat la un laptop. Această opțiune a fost aleasă din motive de simplitate și eficiență, eliminând necesitatea unei surse externe de alimentare în faza de testare și programare. În plus, conexiunea USB servește simultan ca interfață pentru încărcarea codului sursă pe microcontroler, ceea ce reduce complexitatea hardware și accelerează procesul de dezvoltare.

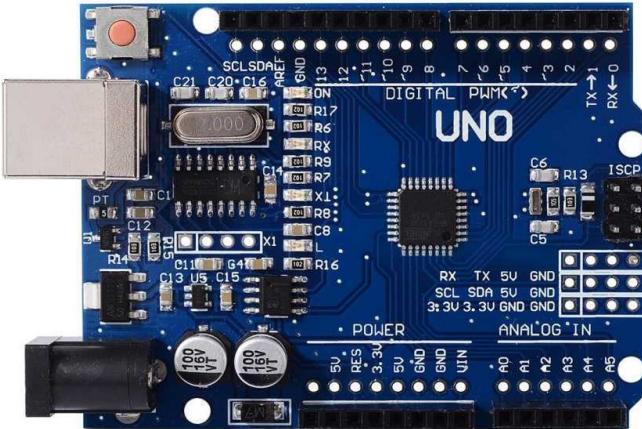


Fig. 3.1.1. - placă UNO

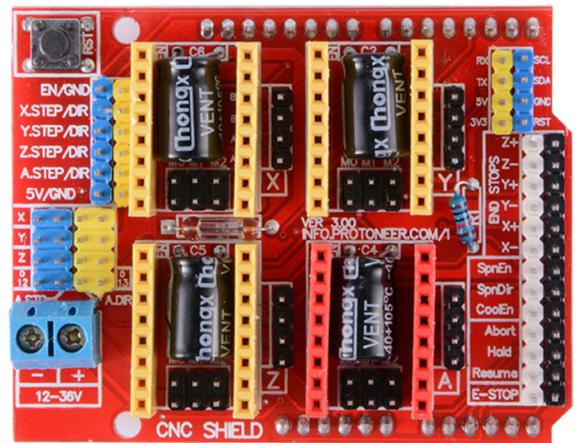
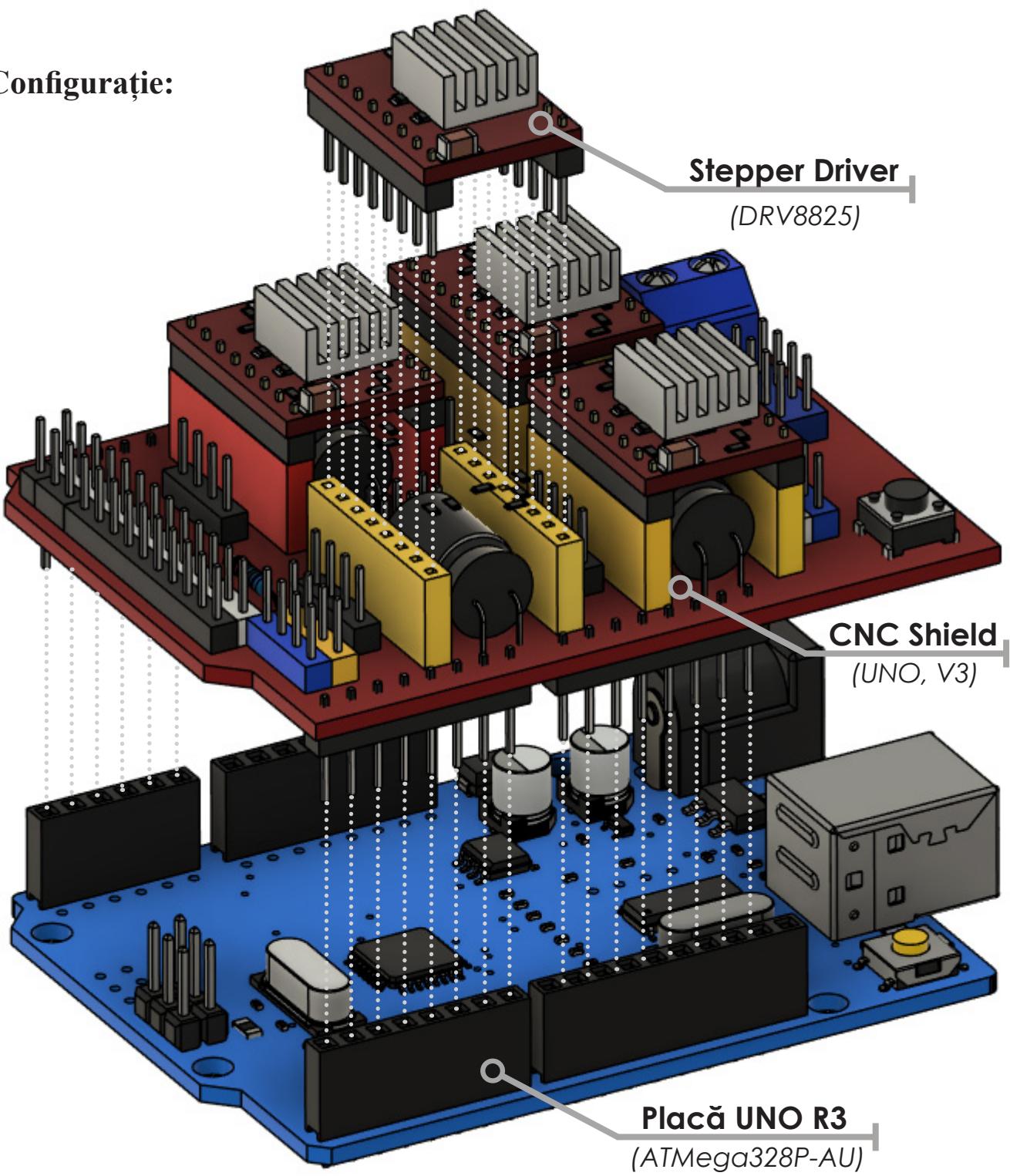


Fig. 3.1.2. - CNC Shield UNO V3

Configurație:



Pentru a garanta siguranța circuitului și pentru a facilita înlocuirea rapidă și eficientă a componentelor auxiliare, am optat pentru utilizarea unui CNC Shield V3, complet compatibil cu placa de dezvoltare Arduino UNO. Alegerea acestui shield a fost determinată de versatilitatea sa și de integrarea facilă a mai multor componente esențiale într-un singur modul, ceea ce reduce semnificativ complexitatea conexiunilor și riscul de erori în montajul circuitului. În plus, CNC Shield V3 oferă o soluție organizată și robustă pentru gestionarea conexiunilor electrice necesare în controlul motoarelor pas cu pas.

Acet shield dispune de patru sloturi dedicate pentru driverele motoarelor pas cu pas, ceea ce îl face ideal pentru proiecte care implică controlul simultan al mai multor axe, cum este cazul robotului nostru. De asemenea, acceptă o gamă largă de tensiuni de alimentare, între 12V și 36V, permitând astfel o flexibilitate mai mare în alegerea sursei de alimentare, în funcție de cerințele de cuplu și viteză ale motoarelor utilizate. Un alt avantaj major îl constituie multitudinea de pini de intrare și ieșire disponibili, fiecare având funcționalități bine definite, care vor fi detaliate în capitolul dedicat software-ului.

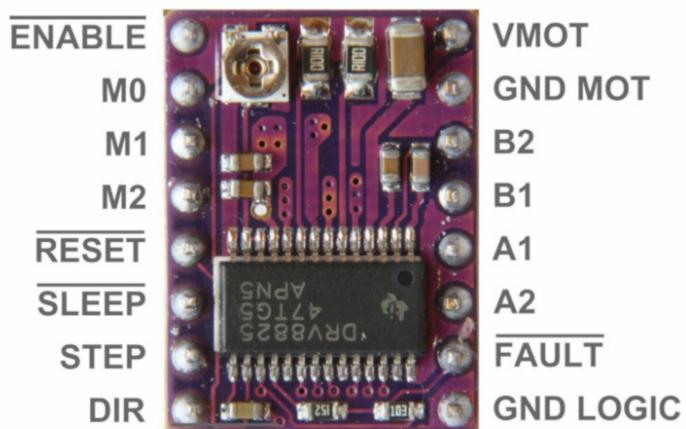


Fig. 3.1.3. - Schema DRV8825

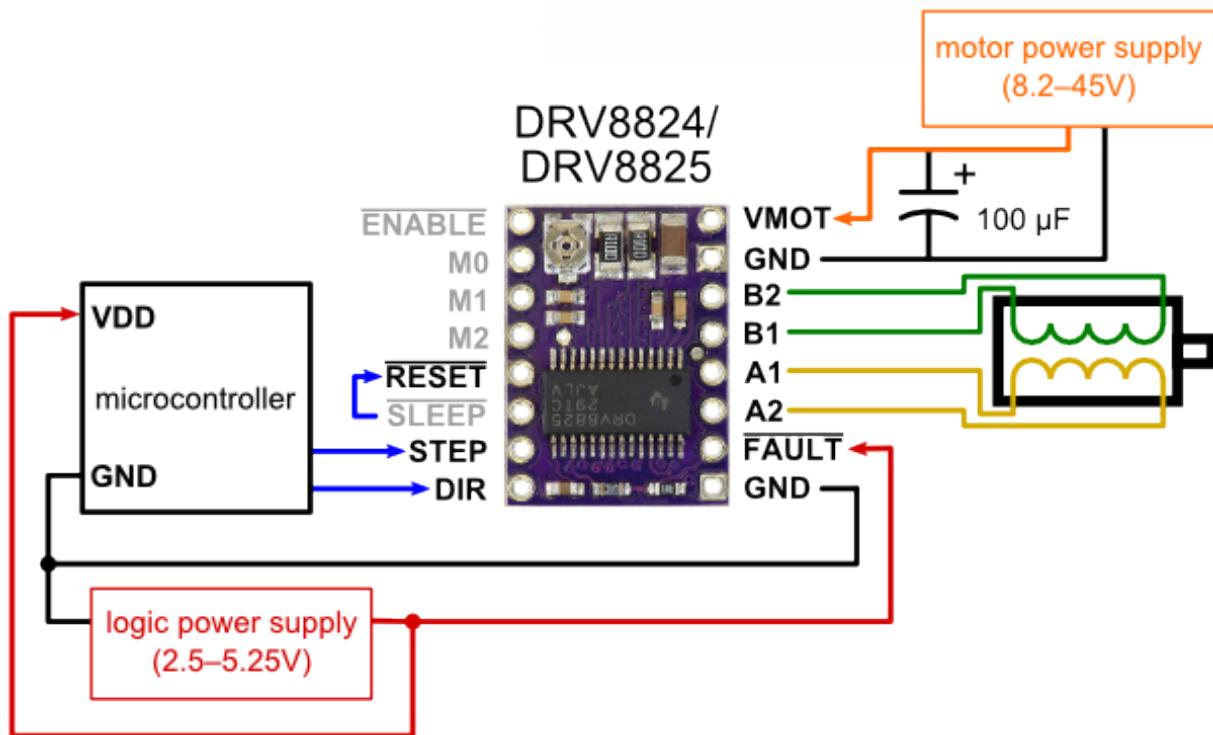


Fig. 3.1.4. - Configurația pinilor DRV8825

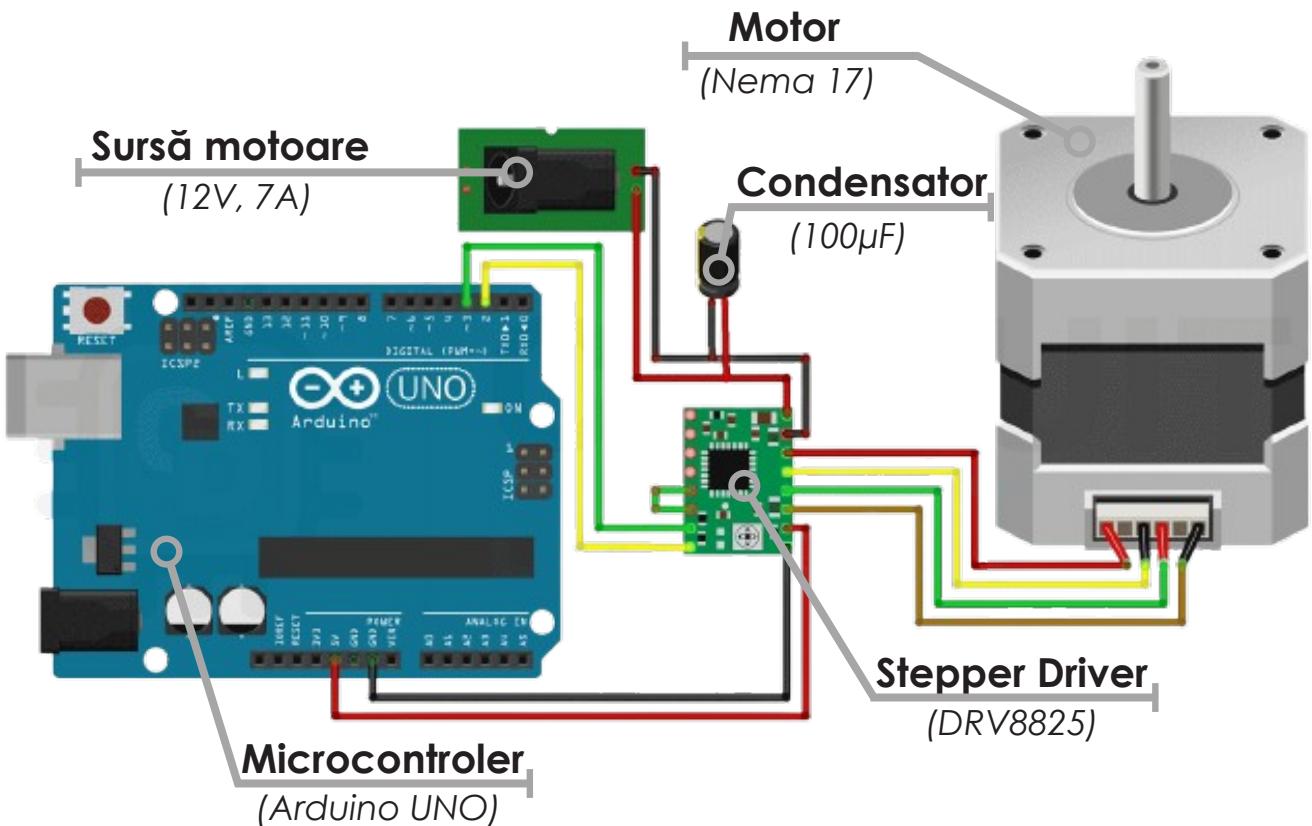


Fig. 3.1.5. - Circuit prototipare schemă

Inițial, pentru a testa funcționalitatea generală a circuitului și pentru a putea face modificări rapide, am conceput o variantă de prototipare utilizând un breadboard. Breadboard-ul este adesea utilizat în fazele incipiente ale dezvoltării, deoarece nu necesită lipire, oferă o modularitate sporită și facilitează testarea diverselor configurații.

Așa cum am menționat anterior, motoarele cu pas sunt alimentate separat, printr-o sursă de 12V, 7A, capabilă să furnizeze curentul necesar pentru a susține cuplul și efortul mecanic al acestora. În paralel, placa de dezvoltare și driverele DRV8825 sunt alimentate prin portul USB al laptopului, o alegere care simplifică procesul de programare și permite o separare clară a circuitelor de putere și de control.

Motivul principal pentru care am renunțat repede la această variantă a fost intensitatea curentului necesară funcționării motoarelor și driverelor, care depășea cu mult capacitatea internă a conexiunilor breadboard-ului. Aceste conexiuni, fiind proiectate pentru curenți mici, au devenit instabile și nesigure, ceea ce poate duce la căderi de tensiune, comportamente imprevizibile sau chiar deteriorarea componentelor.

Așa cum se poate observa și în schița electrică, toate componentele împărtășesc același punct de masă, realizând o împământare comună. Această configurație este esențială pentru stabilitatea sistemului, asigurând referința de tensiune unică necesară funcționării corecte a întregului ansamblu electronic.

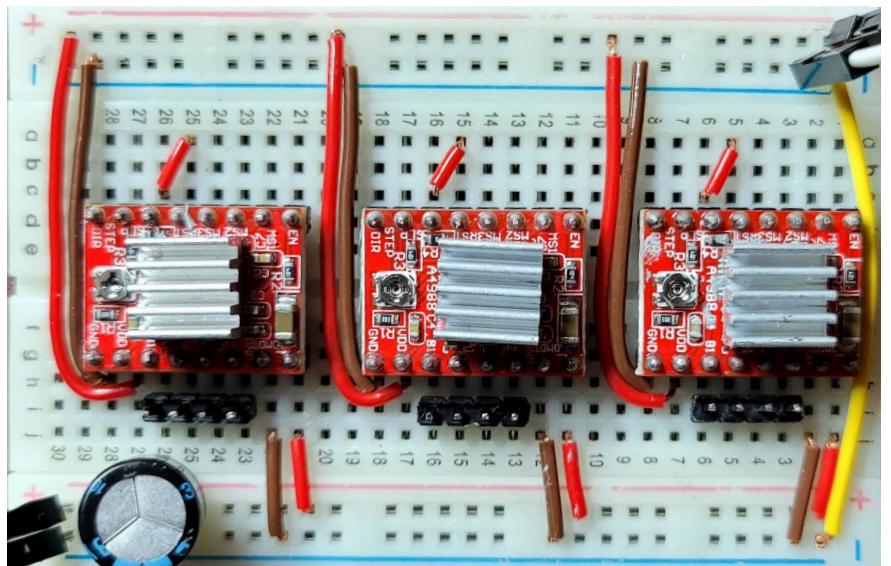


Fig. 3.1.6. - Circuit prototipare breadboard

III.2. Senzori

1) Limit Switch-uri

Așa cum am menționat în capitolele anterioare, pentru inițializarea robotului la poziția 0 este nevoie ca toate brațele să atingă o poziție specifică, convențional aleasă 0. Această facilitate permite folosirea unor valori constante după fiecare utilizare și o putem numi „proces de calibrare”. Din punct de vedere al senzorilor am ales trei switch-uri de tip micro, asemenea celor din imaginea alăturată, capabile să detecteze apăsarea la momentul exact. În cadrul circuitului, senzorii folosec pinii X+, Y+, Z+ ale CNC shield-ului, folosind GND-urile plăcuței și pinii de N/O (normally open) ai senzorilor. Astfel, la momentul contactului, circuitul se închide, iar robotul poate sesiza contactul.

2) Senzor de proximitate

Pentru buna funcționare a sistemului de dozare, este necesar un senzor pentru numărarea capsulelor necesare în umplerea recipientului. În acest scop, am folosit senzorul de proximitate LJ12A3-4-Z/BX, care poate detecta obiecte metalice la până la 4mm distanță. Am folosit acest senzor din motive de accesibilitate, dotând fiecare machetă a capsulelor cu o componentă metalică pentru a putea fi detectată.

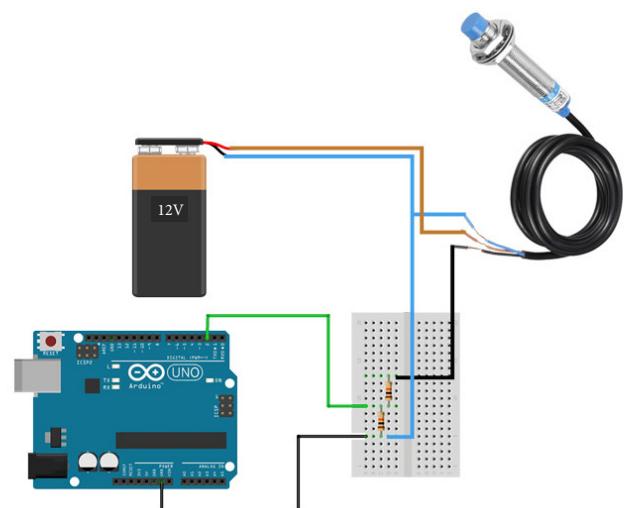
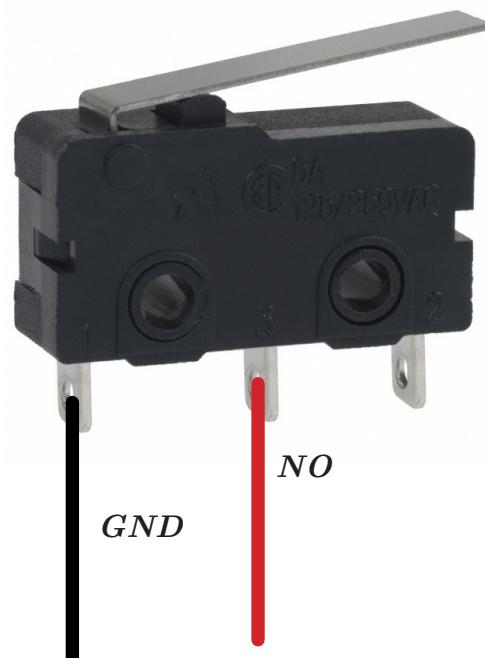
Referitor la montaj, senzorul a fost alimentat din sursa de 7A, 12V, asemenea celorlalte componente active, iar cablul de GND a fost montat atât de minusul sursei, cât și de împământarea plăcuței. Partea de transmitere a semnalului a fost realizată prin întocmirea unui sistem de scădere a tensiunii, prin utilizarea a doi rezistori de 10k ohmi, așa cum se poate observa în figura alăturată. Această scădere de tensiune a fost necesară din cauza output-ului de 12V al senzorului în portul de semnal al arduino-ului (pin analog A0 aka pin 14), care putea face față doar tensiunilor mai mici de 5V. În acest mod, după implementarea circuitului, placă UNO a reușit să detecteze cu succes semnalele senzorului de proximitate.

CALCULE:

Divizorul de tensiune funcționează cu două rezistențe în serie: tensiunea se „împarte” între ele. R_2 este cea legată la GND. Cu cât R_2 este mai mare față de R_1 , cu atât ieșirea e mai aproape de tensiunea de intrare. Cu cât e mai mică, ieșirea scade. Ideal pentru a reduce 12V la sub 5V pentru Arduino.

$$V_{out} = V_{in} \times R_2 / (R_1 + R_2)$$

Pentru $R_1 = R_2 = 10k$ ohm $\Rightarrow V_{out} = 6V$, din motive de accesibilitate, aceasta va fi valoarea temporară, până la introducerea unui alt rezistor.



IV. Software

```
//homming (//hm)
```

Comanda se ocupă de aducerea motoarelor în mediu cunoscut cu ajutorul senzorilor micro de apăsare. Procesul se realizează în mai mulți pași, asigurând poziționarea precisă a brațelor până la finalizarea procesului (x secunde).

```
// Step 1: Fast approach until switch is hit  
stepper.setMaxSpeed(fastSpeed);  
stepper.setSpeed(direction * fastSpeed);  
while (digitalRead(limitPin) == HIGH) {  
    stepper.runSpeed();  
    yield();  
}
```



```
// Step 4: Set position  
stepper.setCurrentPosition(1080);
```

În pasul 1, fiecare braț se apropiie se senzori, având o viteză mai mare. Ulterior, în pașii 3 și 2, robotul se va îndepărta puțin de senzori și se va reapropiă, cu o viteză mult mai mică, pentru a asigura precizia.

La final, în pasul 4, li se atribuie stepperelor noile poziții. Având în vedere că motorul se află la 30 deg față de orizontală după ce este inițializat folosim următoarea formulă pentru conversia poziției.

$$\begin{aligned}stepper.currentposition &= \\&(30 \cdot 1.8 \cdot 5) + (90 \cdot 1.8 \cdot 5) \\&= 1080\end{aligned}$$

Prima paranteză are rolul de a converti cele 30 de grade în pași (1.8deg/step), ținând cont de transmisia motoarelor (5:1). Celelalte 90 deg convertite sunt pentru a simplifica codul, eliminând componenta negativă a poziției. Aceasta valoare va fi adăugată ulterior și tuturor pozițiilor de tip “target” pentru a returna rezultate corecte.

O implementare care a garantat buna funcționare a comenzi a fost verificarea dacă senzorii au semnale LOW la începutul procesului de inițializare. Lipsa acestei structuri de decizie poate duce la avarierea sistemui fizic.

```
//library (//lbr)
```

Comanda afișează în consolă atât toate comenzi disponibile și explicarea lor pe scurt, cât și descrierea unor erori care ar putea apărea în timpul utilizării.

```
void library(){  
    //title  
    Serial.print("-----LIBRARY-----");  
  
    //home  
    Serial.print("//home && /hm");  
    Serial.print(" ->command used for calibrating the position of the robotic arm");
```

```
//newtarget (//nt)
```

Comanda cere introducerea a 3 coordonate (x, y, z), reprezentând poziția finală a platformei efectoare. După introducerea acestor valori, robotul trebuie să calculeze pozițiile exacte ale motoarelor, păstrând constant ca puncte de referință poziția senzorului aferent. După executarea poziționării, robotul așteaptă următoarea comandă.

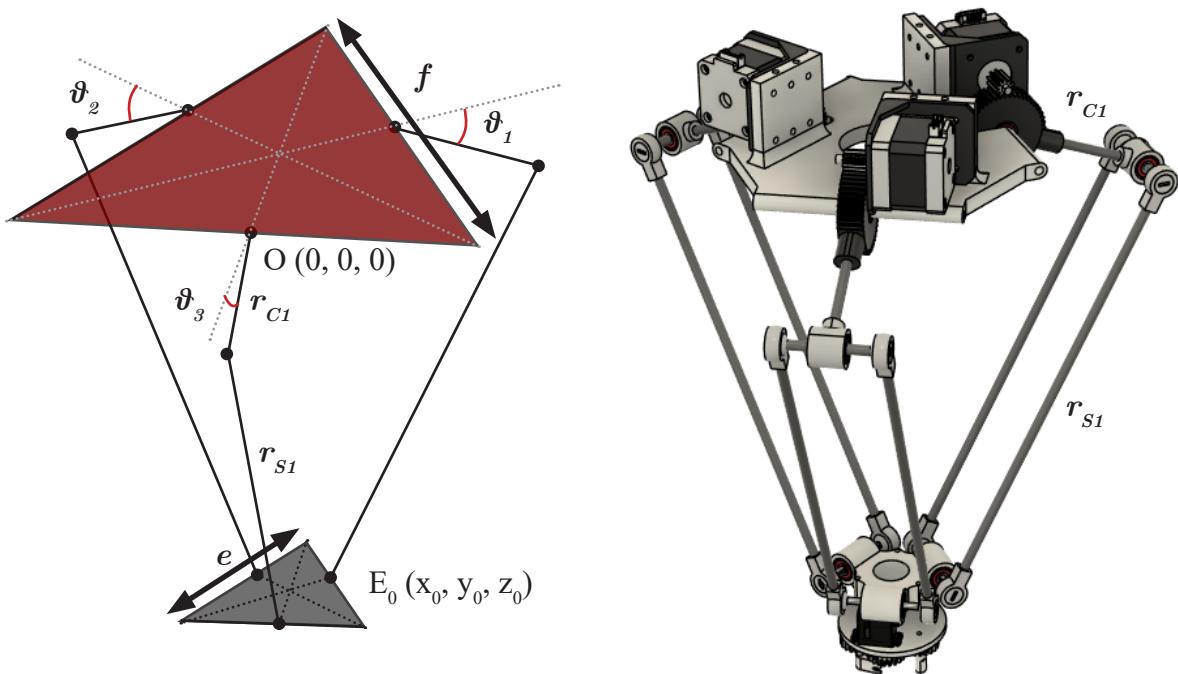
!!!Comanda nu poate fi efectuată fără apelarea comenzi //home în prealabil !!!

Principii de mobilitate

Definirea problemei

Dacă știm poziția dorită a efectostrului final, trebuie să determinăm unghiurile corespunzătoare ale fiecăruiu dintre cele trei brațe (unghiurile articulațiilor) pentru a seta motoarele (și, astfel, efectostrul final) în poziția corectă pentru prindere. Procesul de determinare a acestor unghiuri este cunoscut sub numele de cinematica inversă.

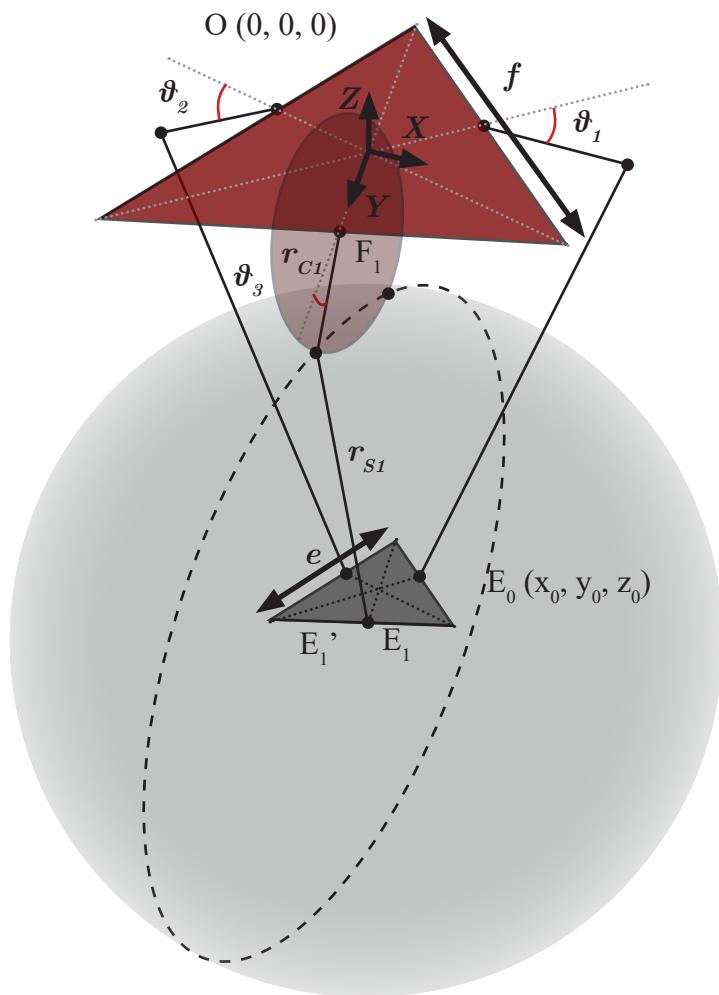
Pentru a fi mai formali, să ne uităm la schema cinematică a robotului. Platformele sunt două triunghiuri echilaterale: cel fix cu motoare este verde, iar cel mobil cu efectostrul final este roz. Unghiurile articulațiilor sunt θ_1 , θ_2 și θ_3 , iar punctul E_0 reprezintă poziția efectostrului final cu coordonatele (x_0, y_0, z_0) . Pentru a rezolva problema cinematică inverse, trebuie să creăm o funcție cu coordonatele lui E_0 (x_0, y_0, z_0) ca parametri, care returnează $(\theta_1, \theta_2, \theta_3)$.



Mai întâi, să determinăm câțiva parametri cheie ai geometriei robotului nostru. Vom nota latura triunghiului fix cu f , latura triunghiului efectostrului final cu e , lungimea brațului superior (al articulației fixe) cu r_p iar lungimea articulației tip paralelogram cu r_e . Aceștia sunt parametri fizici care sunt stabiliți prin proiectarea robotului. Sistemul de referință va fi ales cu originea în centrul de simetrie al triunghiului fix, aşa cum se arată mai jos, astfel că coordonata z a efectostrului final va fi întotdeauna negativă.

Datorită designului robotului, articulația F_1J_1 (vezi figura de mai jos) poate să se rotească doar în planul YZ, formând un cerc cu centrul în punctul F_1 și rază r_f . Spre deosebire de F_1, J_1 și E_1 sunt așa-numite articulații universale, ceea ce înseamnă că legătura E_1J_1 se poate roti liber în jurul lui E_1 , formând o sferă cu centrul în punctul E_1 și rază r_e .

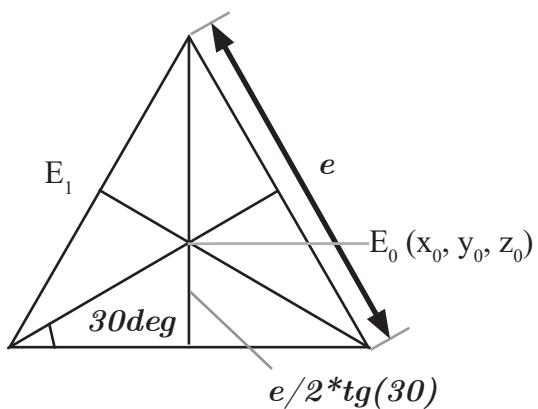
Rezolvare



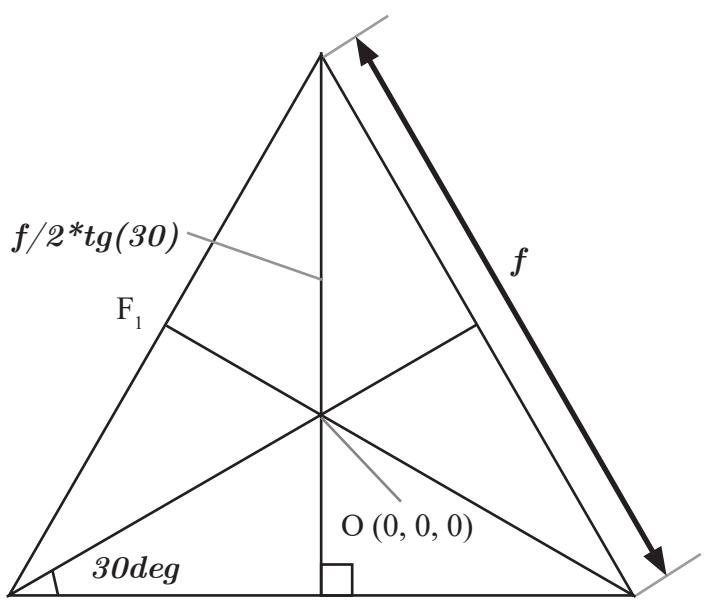
Legendă:

- XZY** - reperul de axe ales;
- O** - originea sistemului, centrul bazei;
- F₁** - îmbinarea dintre bază și brațul superior, centru pinionului cu 50 de dinți;
- J₁** - intersecția între cele două segmente ale brațului;
- E₁** - îmbinarea dintre platformă efectoare și brațul inferior;
- E₁'** - proiecția punctului E₁ pe sferă S₁;
- E₀** - centrul platformei efectoare, aflată la coordonatele alese x_0 , y_0 , z_0 ;
- C₁(F₁, F₁J₁)** - cercul determinat de brațul superior;
- C₂(E₁', E₁'J₁)** - cercul format de proiecția sferei S₁ pe planul YZ;
- S₁(E₁, E₁J₁)** - sferă de terminată de mișcarea brațului inferior;
- θ_{1,2,3}** - unghiurile motoarelor față de axa Y, și rotirile sale în funcție de O la 120, respectiv -120 grade.

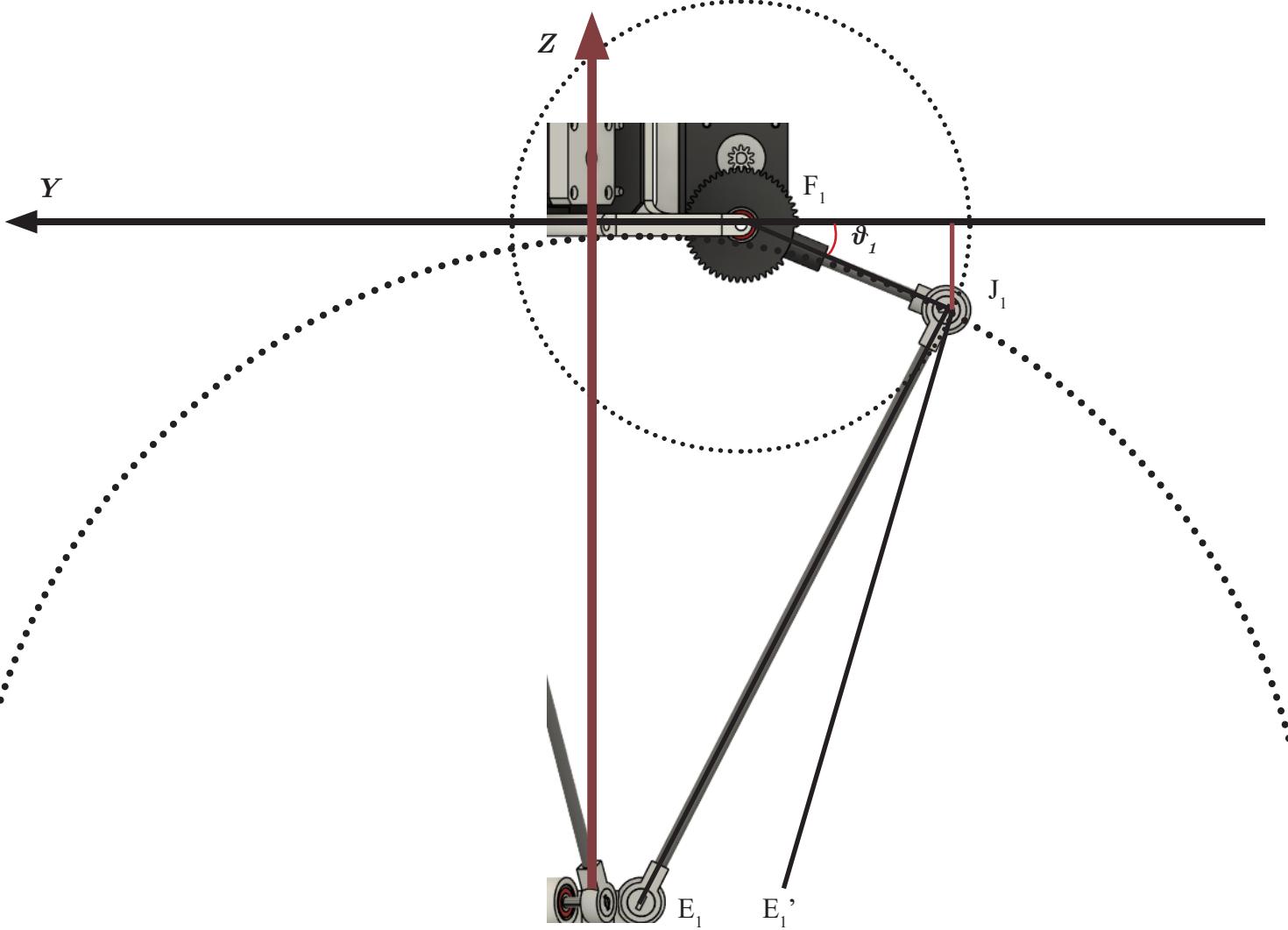
Proiecția sferei determinate de mișcarea brațului inferior pe planul YZ (căruiia îi aparține și C₁) este un cerc cu centrul în E₁' și raza E₁'J₁, unde E₁' este proiecția punctului E₁ pe planul YZ. Așadar, E₁E₁' este perpendicular pe planul XY. După definirea acestor notații și relații, putem observa că punctul J₁ poate fi găsit la una dintre intersecțiile dintre cercurile C₁ și C₂, cu raze cunoscute (este necesar să alegem valoarea din zona de acțiune a robotului, și anume intervalul accesibil). După calcularea coordonatei exacte a punctelor de tip J, putem calcula ușor unghiurile de tip theta. Inițial, ne rezumăm la i=1 pentru toate punctele, folosim planul XY, după care rotim planul pentru determinarea valorilor corespondente i=2 (cu 120 deg) și i=3 (cu -120 deg).



Platformă efectoare



Bază



$E_0(x_0, y_0, z_0)$ este punctul de pornire al rezolvării, presupunând că cunoaștem poziția platformei efectoare, deci x_0, y_0, z_0 cunoscute.

$$\begin{aligned} E_1 E &= e/2 \cdot \tan(30) = e/(2 \cdot \sqrt{3}) \\ \text{unde } e &= \text{latura platformei} \\ F_1 O &= f/2 \cdot \tan(30) = f/(2 \cdot \sqrt{3}) \\ \text{unde } f &= \text{latura bazei} \end{aligned} \quad \Rightarrow \quad \begin{aligned} E_1(x_0, y_0 - e/(2 \cdot \sqrt{3}), z_0) \\ E_1'(0, y_0 - e/(2 \cdot \sqrt{3}), z_0) \\ F_1(x_0, -f/(2 \cdot \sqrt{3}), z_0) \end{aligned}$$

$$\Rightarrow E_1 E_1' = x_0 \quad \Rightarrow \quad J_1 E_1' = \sqrt{(J_1 E_1'^2 - E_1 E_1'^2)} = \sqrt{r_{S1}^2 - x_0^2}$$

Obținem următorul sistem de ecuații care caracterizează punctele posibile J_1 :

$$\left\{ \begin{array}{l} (y_{J1} - y_{F1})^2 + (z_{J1} - z_{F1})^2 = r_{C1}^2 \\ (y_{J1} - y_{E1})^2 + (z_{J1} - z_{E1})^2 = r_{S1}^2 - x_0^2 \end{array} \right. \iff \left\{ \begin{array}{l} (y_{J1} + f/(2 \cdot \sqrt{3}))^2 + z_{J1}^2 = r_{C1}^2 \\ (y_{J1} - y_0 + e/(2 \cdot \sqrt{3}))^2 + (z_{J1} - z_0)^2 = r_{S1}^2 - x_0^2 \end{array} \right.$$

$$\Rightarrow J_1(0, y_{J1}, z_{J1}) \quad \Rightarrow \quad \vartheta_1 = \arctan(z_{J1}/(y_{F1} - y_{J1}))$$

Această simplificare a calculelor permite omiterea totală a axei X. Pentru un rezultat cât mai direct și orecesis, profităm de simetria robotului pentru theta 2 și theta 3. Așadar, vom roti sistemul de coordonate în jurul axei Z cu -120 și 120 de grade.

Rezultă că:

pentru *theta 1* vom folosi coordonatele (x_0, y_0, z_0)
 pentru *theta 2* vom folosi coordonatele $(x_0 * \cos 120 + y_0 * \sin 120, y_0 * \cos 120 - x_0 * \sin 120, z_0)$
 pentru *theta 3* vom folosi coordonatele $(x_0 * \cos 120 - y_0 * \sin 120, y_0 * \cos 120 + x_0 * \sin 120, z_0)$

Implementare



```
int delta_calcAngleYZ(float x0, float y0, float z0, float &theta) {
    float y1 = -0.5 * 0.57735 * f; // f/2 * tg 30
    y0 -= 0.5 * 0.57735 * e; // shift center to edge
    // z = a + b*y
    float a = (x0*x0 + y0*y0 + z0*z0 + rf*rf - re*re - y1*y1)/(2*z0);
    float b = (y1-y0)/z0;
    // discriminant
    float d = -(a+b*y1)*(a+b*y1)+rf*(b*b*rf+rf);
    if (d < 0) return -1; // non-existing point
    float yj = (y1 - a*b - sqrt(d))/(b*b + 1); // choosing outer point
    float zj = a + b*yj;
    theta = 180.0*atan(-zj/(y1 - yj))/pi + ((yj>y1)?180.0:0.0);
    return 0;
}
```

Implementarea transpună calculul matematic efectuat pentru calcularea unghiurilor theta într-o funcție de tip int, care returnează unghiul căutat. Parametrii de intrare sunt cele trei coordonate x, y, z (diferire în funcție de poziționarea fiecărei îmbinări superioare), iar theta este parametrul de ieșire. Pentru a ne asigura că J ia valoarea corectă dintre cele două posibile, o vom alege pe cea mai departată de centrul bazei robotului.



```
Targetx=-theta1*1.8*5+810;
Serial.println("Targetx = " + String(Targetx, 4));
```

După aflarea unghiurilor, convertim unghiurile în noua poziție vizată de motoare (în pași), folosind formula elaborată anterior.

După stabilirea poziției vizate, se începe procesul de mișcare. În primul rând, se stabilește sensul motorului raportat la poziția inițială. Ulterior, se produce mișcarea spre noua valoare și reinicializarea motorului la poziția curentă.

```
int pullOffStepsX=abs(stepperX.currentPosition()-Targetx);
if(stepperX.currentPosition()>Targetx){
    stepperX.setSpeed(200);
} else
    stepperX.setSpeed(-200);

for (int i = 0; i < pullOffStepsX; i++) {
    stepperX.runSpeed();
    delay(1);
    yield();
}

stepperX.setCurrentPosition(1000);
```

```
//count (/ct)
```

Funcția count se ocupă exclusiv de coordonarea și punerea în mișcare a sistemului de dozare cu care este prevăzut robotul, selectând exact numărul dorit de capsule pentru umplerea recipientului.

Inițial, programul pune în mișcare discul inferior al mecanismului, păstrând această mișcare de rotație cât timp numărul dorit de pastile nu a fost atins. Odată ce capsulele își încep mișcarea pe canalul lateral și trec pe sub senzorul de proximitate, acesta începe să facă numărarea.

Contraște așteptărilor, procesul de numărare nu verifică numărul de valori de tip True pe care îl recepționează senzorul, deoarece viteza capsulelor este imprevizibilă, iar o oprire neașteptată a unei capsule sub senzor ar putea duce la erori semnificative. Așadar, în locul acestei verificări, vom calcula numărul de interschimbări True-False, ci nu de câte ori apar acestea. Modelul False- True - False (în diferent de numărul de dubluri a fiecarei comenzi la momentul recepționării) indică, în mod evident, numărarea unei capsule. Funcția utilizată în cazul nostru este elaborată în următorul cod:

```
void dispenser() {
    int detectionCount = 0;
    int lastSensorState = digitalRead(SENSOR_PIN);

    stepperE.setSpeed(125); // Set speed for continuous motion
    Serial.println("Dispensing started...");

    while (detectionCount < 5) {
        stepperE.runSpeed(); // Keeps motor moving

        int currentSensorState = digitalRead(SENSOR_PIN);

        // Count falling edge (object appears)
        if (currentSensorState == LOW && lastSensorState == HIGH) {
            detectionCount++;
            Serial.print("Object Detected #");
            Serial.println(detectionCount);
        }

        lastSensorState = currentSensorState;
    }

    stepperE.setSpeed(0); // Stop motor
    Serial.println("Dispensing complete.");
}
```

După obținerea numărului dorit de capsule, discul inferior este oprit, prin urmare și cel inferior, oprind dozarea și anunțând finalul operațiunii. În codul de mai sus este ilustrată funcția “dispenser”, apelată anterior în loop, după introducerea comenзii /count.

//catch (//ch), respectiv //release (//rl)

Comenziile se ocupă de închiderea și deschiderea ghearei de preluare, acționând conform unor poziții prestabilite.

Bibliografie

Pentru proiectarea robotului am folosit programul Autodesk Fusion 360 (<https://www.autodesk.com/products/fusion-360/overview>), pe care l-am importat ulterior în Blender (<https://www.blender.org/download/>) și Unity (<https://unity.com>) pentru simulări. Referitor la programare, am folosit Arduino IDE (<https://www.arduino.cc/en/software/>), iar pentru simulări geometrice în plan am apelat la GeoGebra (<https://www.geogebra.org>). Redactarea acestei lucrări a fost făcută în Word (<https://word.cloud.microsoft>), iar unele dintre imagini în InDesign (<https://www.adobe.com/ro/products/indesign.html>).

1. Hugo Hadfield , Lai Wei , Joan Lasenby, The Forward and Inverse Kinematics of a Delta Robot, University of Cambridge, Department of Engineering, Cambridge, UK

<https://www.repository.cam.ac.uk/items/b080ea69-d294-4f7b-99c2-39cc64415f57>

2. Trossen Robotics Community, Delta robot kinematics

<https://hypertriangle.com/~alex/delta-robot-tutorial/>

3. Marginally Clever Robots, Rotary Delta Robot Forward/Inverse Kinematics Calculations

<https://www.marginallyclever.com/other/samples/fk-ik-test.html>

4. Robert L. Williams II, The Delta Parallel Robot: Kinematics Solutions, Mechanical Engineering, Ohio University, October 2016

<https://people.ohio.edu/williams/html/PDF/DeltaKin.pdf>