Proiect component: Pr.5 *Conducerea inteligentă, cu tehnici avansate şi navigatia bazată pe senzori performanţi, sistem video-biometric şi sistem servoing vizual a sistemului autonom complex SAC-SI integrat în tehnologia de asistare a persoanelor cu dizabilităţi neuro-motorii severe*

Activitatea: Act 2.13 - *-Proiectarea structurii de conducere inteligentă (bazată pe tehnici avansate) precum şi a structurii de navigaţie (bazată pe senzori performanţi) pentru sistemul autonom complex SAC-SI integrat in tehnologia de asistare a persoanelor cu dizabilitati neuro-motorii severe*

Indicatori de realizare:
-Structură de conducere inteligentă si structură de navigare pentru SAC-SI, "Cirrus Power Wheelchair" integrat în tehnologia de asistare a persoanelor cu dizabilităţi neuro-motorii severe;
-Produs software nou

În cadrul acestei activităţi s-a dezvoltat o metoda de comandă a unui scaun cu rotile pentru persoanele cu dizabilităţi severe care nu pot acţiona manual joystick-ul scaunului cu rotile.

S-a pornit de la ideea îmbunătăţirii unui scaun cu rotile, dotat cu motoare electric şi cu echipamentul de calcul necesar pentru a efectua detecţia feţei şi pentru a trimite comenzile către motoare, scopul final fiind controlul simplu, sigur şi uşor de învăţat pentru o persoana cu dizabilităţi locomotorii severe.

Folosind metoda propusa, scaunul cu rotile poate fi controlat astfel încât acesta poate merge înainte, poate vira, şi poate merge cu viteză variabilă în funcţie de nevoile utilizatorului.

Schema general propusă este prezentată în Figura 2.13.1.

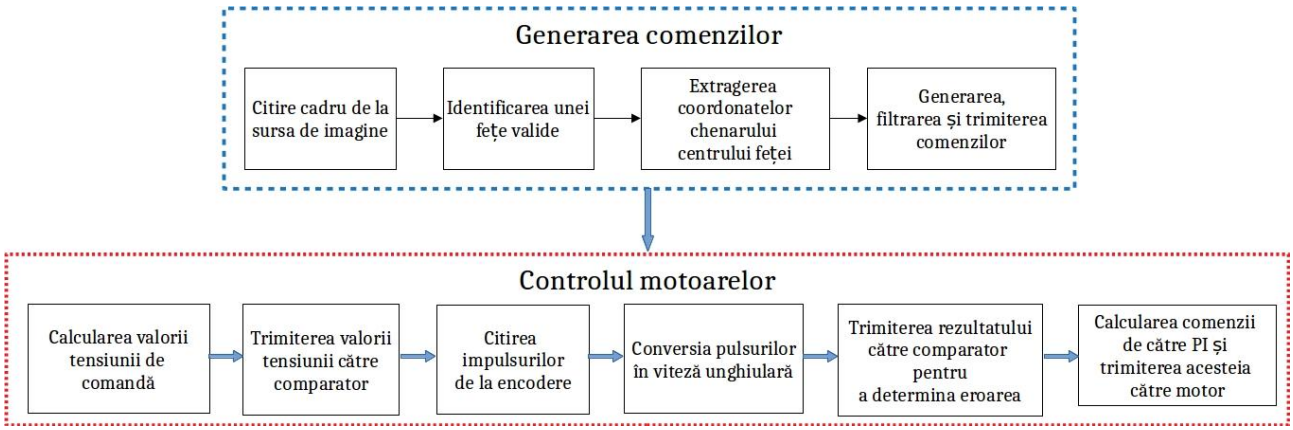

Fig. 2.13.1 Schema bloc funcţională.

Pentru a putea efectua detecţia feţei în timp real s-a utilizat librăria OpenCV (OpenCV este o bibliotecă software, open-source ce are o multitudine de utilizări în domenii ce necesită recunoaşterea formelor obiectelor). În proiectul de faţă s-a utilizat o camera Microsoft LifeCam HD-3000 care are o rezoluţie de 1280x720 pixeli şi care furnizează 30 de cadre pe secundă.

Fiecare cadru achiziţionat de către cameră trebuie prelucrat pentru a permite clasificatorului din OpenCV să identifice faţa utilizatorului. Prelucrarea ce are loc imediat după achiziţia fiecărui cadru este convertirea cadrului color într-unul alb-negru. După ce imaginea color a fost transformată într-o imagine alb-negru, aceasta este trimisă către un clasificator care este responsabil de detecţia propriu-zisă a feţei.

În cazul de faţă, clasificatorul este o componentă software care foloseşte un set de reguli pentru identificarea feţei utilizatorului. Setul de reguli pentru clasificator este obţinut prin introducerea unor imagini ce reprezintă exemple pozitive (imagini în care apar feţe umane) şi negative (imagini fără feţe) într-o aplicaţie ce utilizează toate eşantioanele introduse pentru a determina ce anume doreşte utilizatorul să fie identificat şi ce nu.

Clasificatorul folosit în această lucrare este Haar Cascade. Acesta primeşte imaginea alb-negru procesată anterior şi începe sa caute în imagine caracteristici (forme) ce corespund setului de reguli obţinut anterior utilizând o fereastră de căutare.

Fereastra de căutare joacă un rol foarte important în acurateţea detecţiei feţei. Cu cât fereastra de căutare este mai mică, cu atât rezultatele vor fi mai precise dar nu se va mai putea asigura caracteristica de timp real deoarece acest proces necesită foarte multe resurse hardware şi un timp de calcul ridicat. De aceea trebuie găsită o dimensiune a ferestrei de căutare care să permită o detecţie bună a feţei şi, în acelaşi timp, sa nu consume toate resursele hardware, deoarece în sistem există şi alte procese care au nevoie de timp de calcul.

Pentru generarea comenzilor utilizatorul trebuie sa îşi mişte capul, de exemplu ridicarea uşoară a capului reprezintă comanda „înainte", iar coborârea capului opreşte scaunul cu rotile.

Soluţia aleasa a fost utilizarea a doua patrulatere. Un patrulater, cel iniţial, este utilizat ca şi punct de plecare, iar cel de-al doilea este utilizat pentru determinarea poziţiei actuale a feţei (a se vedea Figura 2.13.2).

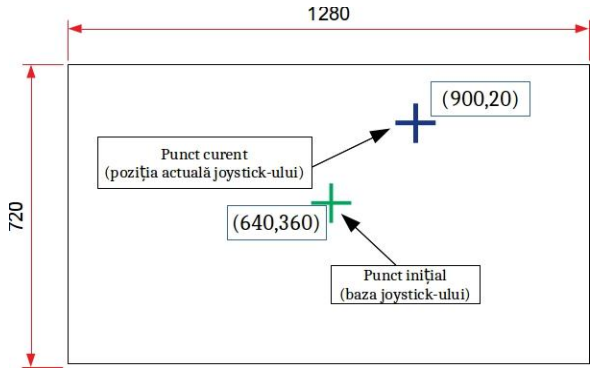

Fig. 2.13.2. Reprezentarea centrele patrulaterelor a căror coordonate sunt utilizate pentru a determina direcţia de mers şi viteza de deplasare.

Fig. 2.13.3. Exemplu de patrulater valid - care încadrează atât faţa utilizatorului cât şi ochii acestuia

În cazul de faţă, un patrulater valid ce poate fi folosit pentru a genera comenzi este un patrulater care încadrează atât faţa utilizatorului cât şi ochii acestuia (a se vedea Figura 2.13.3).

Pentru aceasta se folosesc doua clasificatoare. Un clasificator este pentru faţă, iar celălalt este doar pentru ochi. Dacă patrulaterul pentru ochi este înscris în patrulaterul feţei atunci patrulaterul feţei este considerat valid şi astfel se poate genera următoarea comandă. În caz contrar acesta este ignorat şi se repetă comanda anterioară. Dacă în decurs de 2 secunde nu este detectat nici un patrulater valid atunci scaunul cu rotile va fi oprit din motive de siguranţă, iar utilizatorul va trebui sa plaseze faţa în punctul iniţial după care scaunul cu rotile va putea folosit în continuare.

Pentru generarea comenzii se folosesc doar centrele celor două patrulatere (cel de referinţă şi cel actual), restul atributelor nu mai sunt necesare.

Pentru eliminarea perturbaţii s-a folosit un filtru Kalman. Motivul pentru care s-a ales filtrul Kalman, în detrimentul altor metode de eliminare a perturbaţiilor, este consumul redus de resurse pentru filtrare şi simplitatea implementării. În acelaşi timp, filtrul Kalman oferă o filtrare buna pentru obţinerea unor valori cât mai stabile ale semnalului.

La o primă vedere, modulul bazat pe OpenCV nu pare a fi un sistem cu buclă închisa însă acest lucru nu este adevărat, elementul responsabil de închiderea buclei de control fiind utilizatorul. Acesta joacă rolul de regulator şi de comparator fiind responsabil de „calcularea" erorii (unde vrea să se deplaseze şi unde se duce scaunul cu rotile) precum şi de „generarea" unei comenzi corective.

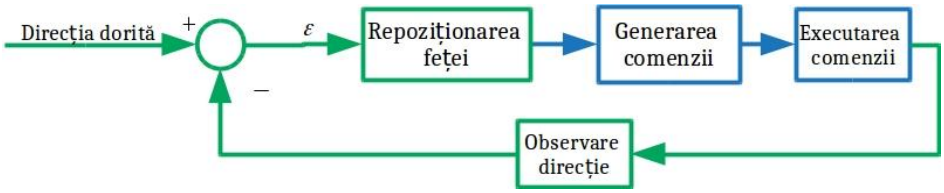

Fig. 2.13.4. Schema de generare a comenzilor în buclă închisă

În Figura 2.13.4 este ilustrat modul în care comanda generată de către utilizator este corectată în cazul în care aceasta nu corespunde cu referinţa (direcţia dorită).

In Figura 2.13.5 este dezvoltată schema algoritmului propus pentru generarea comenzilor.

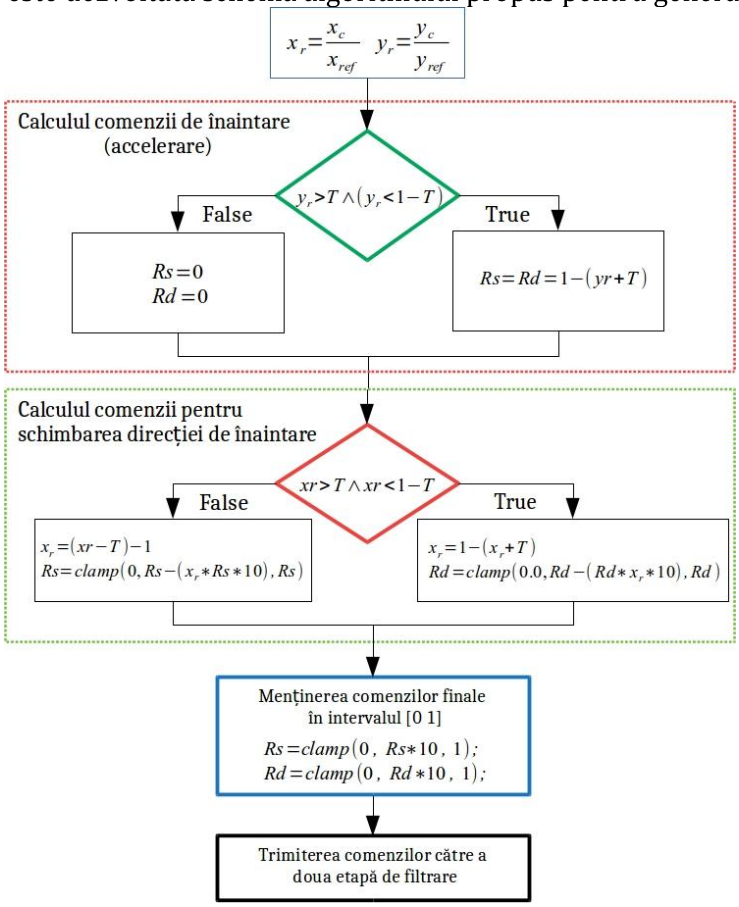

Fig.2.13.5. Schema bloc a algoritmului propus pentru generarea comenzilor

Interfaţa grafică implementată in OpenCV este exemplificată în Figura 2.13.6.

Fig. 2.13.6. Interfaţa grafică a aplicaţiei dezvoltate (produs software nou)

Aplicația realizata a fost proiectata și implementata astfel încât aceasta să poată fi extinsă ușor pentru ca în viitor sa își găsească menirea în a ușura deplasarea persoanelor cu dizabilități și pe alte tipuri de platforme mobile.

Pe baza soluției propuse a fost publicat un articol indexat BDI (cu propunere de indexare ISI Proceedings).

Activitatea: Act 2.14 - *-Proiectarea structurii de evitare a obstacolelor (bazată pe senzori de tip laser si video) pentru sistemul complex SAC-SI integrat în tehnologia de asistare a persoanelor cu dizabilități neuro-motorii severe*

Indicatori de realizare:
-Structură de evitare a obstacolelor (bazată pe senzori de tip laser și video) ce deserveste scaunul cu rotile de tip "Cirrus Power Wheelchair" (SAC-SI) integrat în tehnologia de asistare a persoanelor cu dizabilități neuro-motorii severe;
-Produs software nou;

În cadrul acestei etape s-a realizarea în MATLAB un algoritm pentru determinarea automată a unei traiectorii (ce permite evitarea obstacolelor fixe) pentru un scaun cu rotile/fotoliu rulant cu două roți motoare utilizând PSO (Particle Swarm Optimization).

Fiecare particulă actualizează poziția și viteza sa utilizând următoarele ecuații:

(2.14.1)

Forma generală a funcției fitness este:

(2.14.2)

În metode globale de găsire a unui traseu, mediul se presupune a fi complet cunoscut și traseul (calea) este optimizat luând în considerare toate informațiile care apar în hartă. Traseul de referință poate duce robotul la destinație într-un mod rafinat. Mai mult, traseul optim este găsit numai dacă obstacolele din cale sunt fixe și nu există zone fără ieșire.

Algoritmul utilizat pentru metoda globală este descris în tabelul de mai jos:

Tabelul I: Algoritmul „Path Planning" utilizat pentru metoda globală

```
Initialization Parameters:
  iterations, swarm size, initial swarm position, initial velocity.
FOR  iter = 1 to iterations DO
        FOR i = 1 to swarm size DO
                update particles Si (see eq. 2.14.1)
                calculate the Fitness (see eq. 2.14.2)
                IF new position is better THEN
                        update best Si
                        update the best value
                End IF
        End FOR
        update global best positioning
        FOR i = 1 to swarm size DO
                updating velocity vectors Vi (see eq. 2.14.1)
        End FOR
End FOR
```

In Fig. 2.14.1 este prezentat un studiu de caz cu trei obstacole fixe pe calea robot mobil. Este uşor de observat din această figură că drumul până la destinaţie este găsit după 21 de iteraţii ale algoritmului. Tot în Figura 1 este arătată detaliat situaţia în care robotul trebuie să treacă printre două obstacole fixe. Se poate observa că doar 13 din cele 25 de particule (dimensiunea roiului utilizat pentru acest studiu de caz) sunt luate în considerare. Celelalte particule se suprapun cu obstacolele (11 particule se suprapun peste obstacolul nr. 1 şi o particulă se suprapune peste obstacolul nr. 3).

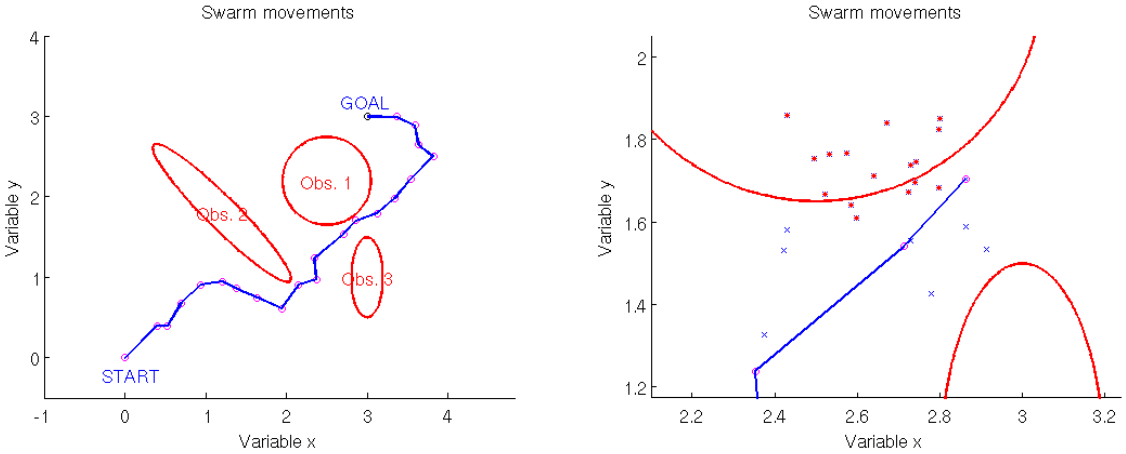

Fig. 2.14.1. Rezultatul algoritmului de găsire a traseului optim între punctul de START şi cel FINAL folosind PSO (metoda globală); detaliul zonei dintre obstacolele 1 şi 3.

Rezultatul final pentru acest algoritm este dat de setul de puncte intermediare ale traseului (x_i, y_i). Setul de puncte intermediare reprezintă intrarea pentru controllerul de urmărire al traseului robot mobil. Sarcina controllerului este de a deplasa autonom robotul mobil pe traseul găsit utilizând algoritmul PSO.

Tot în această etapă s-a implementat in MATLAB un algoritm pentru realizarea automata a unei traiectorii ce ţine cont de restricţiile de timp şi confort pentru un scaun cu rotile. In aceasta etapa au fost generaţi vectorii de viteza (liniara si unghiulara) necesari conducerii in timp real a platformei mobile (sistemului autonom).

Pentru determinarea traiectoriei unui robot mobil se utilizează ecuaţii de gradul 5.

( ) [ ] [ ] (2.14. 3)

Ecuaţia curburii realizată pentru fiecare segment din traiectorie este:

( ) ( ) ( )

(( ) ( )) (2.14.4)

Lungimea curburii este dată de ecuaţia:

( ) ∫ √ ( ) ( )

(2.14.5)

Profilul de viteză care corespunde fiecărui segment este împărţit în 5 părţi m = 1, 2, ..., 5:

( ) ( )

(2.14.6)

Similar, lungimea şi acceleraţia longitudinală pentru fiecare segment sunt calculate astfel:

( )
( )

( ) (2.14.7)
( )

(2.14.8)

Viteza ( ) unghiulară poate fi calculată utilizând formula lui Frenet:

(2.14.9)

iar acceleraţia laterală cu relaţia: ( )

(2.14.10)

Confortul uman este calculat conform ecuaţiei de mai jos (conform ISO 2631-1 şi ISO2631-5)

√ (2.14.11)

unde awx, awy, awz sunt acceletaţiile medii pătratice calculate pe fiecare segment al traiectoriei, iar τ_x, τ_y, τ_z factori de multiplicare: $\tau_x = 1.4$, $\tau_y = 1.4$, $\tau_z = 0$.

Algoritmul propus pentru calcularea vitezelor lineare şi unghiulare ce ţine cont de restricţiile de timp pentru un robot mobil cu două roţi motoare este descris în tabelul II.

Tabelul II: Trajectory Planning Algoritm

1: Calculate $x_{j,j+1}(u)$ and $y_{j,j+1}(u)$ for each curve (eq. 2.14.3)
calculate the curvature $K_{j,j+1}(u)$ for each curve (eq. 2.14.4) and
calculate the curve length $L_{j,j+1}(u)$ for each curve (eq. 2.14.5)
2: Determine a time $t_{j,j+1}$ for each seg. $Pos_{j,j+1}$ of the path
The time is calculate function of the comfort of human
body constraint: $t_{j,j+1} = ((2 \cdot L_{j,j+1})/a_w)^{1/2}$

3: **FOR** j = 1 to goal (final position) **DO**
4: Calculate an average velocity $_{j,j+1} = a_w \cdot t_{j,j+1}$
5: Calculate an initial velocity profile $v_{j,j+1}(t)$ with $t \in [0, t_{j,j+1}]$ (see eq. 2.14.6)
and longitudinal accel. profile $a_{T_{j,j+1}}(t)$ using eq. 2.14.8
6: Calculate angular velocity profile $\omega_{j,j+1}(t)$ (see eq. 2.14.9) and lateral accel.
profile $a_{L,j,j+1}(t)$ using eq. 2.14.10
7: Calculate the overall r.m.s. acc. $a_{wj,j+1}$ for segment (eq. 2.14.11)
8: **IF** $a_{wj,j+1} > 0.4$ m/s^2 **THEN**
increase the time $t_{j,j+1}$ for this segm. and go to step 4
10: **End FOR**

În figurile următoare sunt reprezentate rezultatele obţinute în urma simulării algoritmului propus (tabel II):

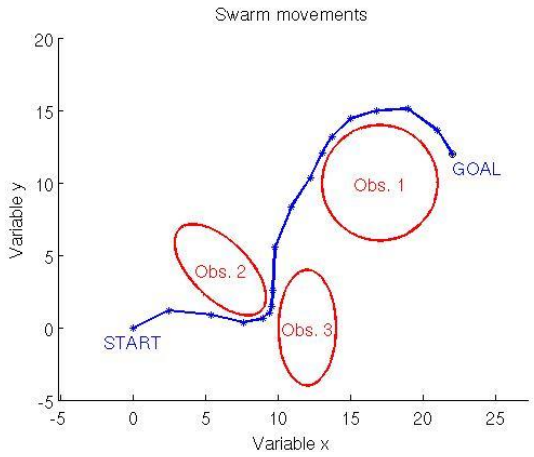

Fig. 2.14.3 Exemplu de traiectorie utilizand ecuaţii de gradul cinci.

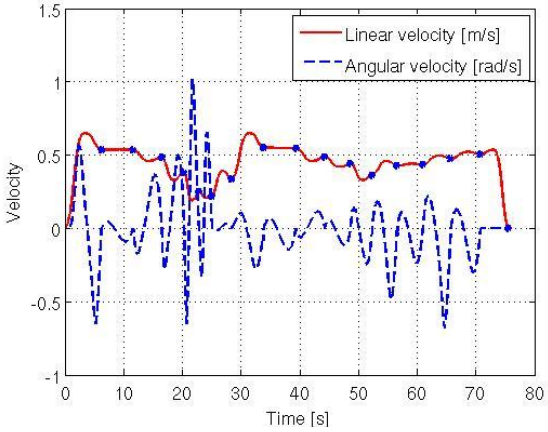

Fig.2.14.4 Viteza liniară şi unghiulară calculate cu ajutorul algoritmului propus.

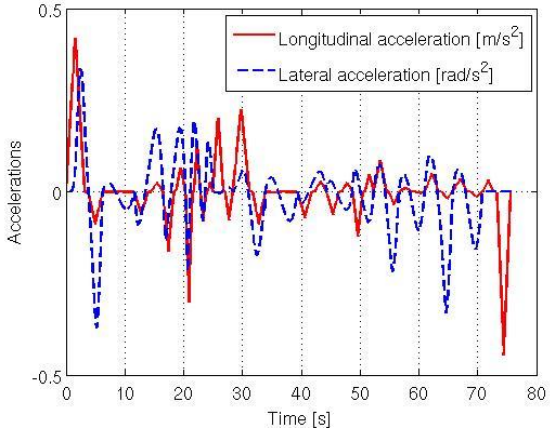

Fig. 2.14.5 Acceleraţiile liniare şi laterale pentru exemplul din figura 2.14.3.

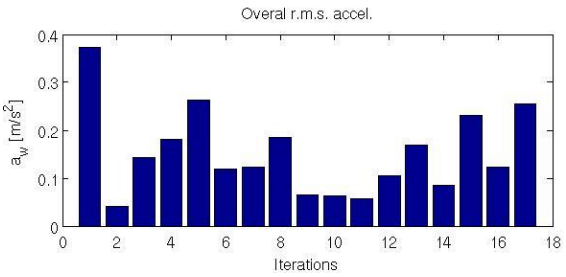

Fig. 2.14.6 Valorile medii pătratice ale acceleraţiilor pentru exemplul din figura 2.14.3.

Activitatea: Act 2.15 - *-Proiectarea şi realizarea structurii de conducere avansată bazată pe sisteme servoing vizuale (pt. manipulatorul robotic cu 7DOF) ce deserveste sistemului autonom complex SAC-SI integrat în tehnologia de asistare a pers. cu dizabilităţi neuro-motorii severe*

Indicatori de realizare:
-Structură de conducere avansată bazată pe senzori performanţi şi sistem servoing vizual mobil a manipulatorului Cyton 1500 care echipează SAC-SI;
-Produs software nou;

In cadrul acestei activitaţi s-a realizat sortarea unor obiecte de pe o suprafaţă de lucru după mărimea lor, urmată de manipularea lor utilizand un brat robotic (Cyton Gamma 1500).

Algoritmul propus oferă posibilitatea selecţiei celui mai mic obiect de pe suprafaţa de lucru, selecţia celui mai mare obiect sau aranjarea obiectelor într-o stivă în ordinea mărimii lor, de la cel mai mare la cel mai mic.

Algoritmul este dezvoltat utilizând software-ul MATLAB, împreuna cu toolbox-ul pentru procesare de imagini. Detectarea obiectelor este o tehnologie care face parte din domeniul procesării computerizate a imaginilor.

Structura echipamentelor hardware utilizate este prezentată în figura 2.15.1.

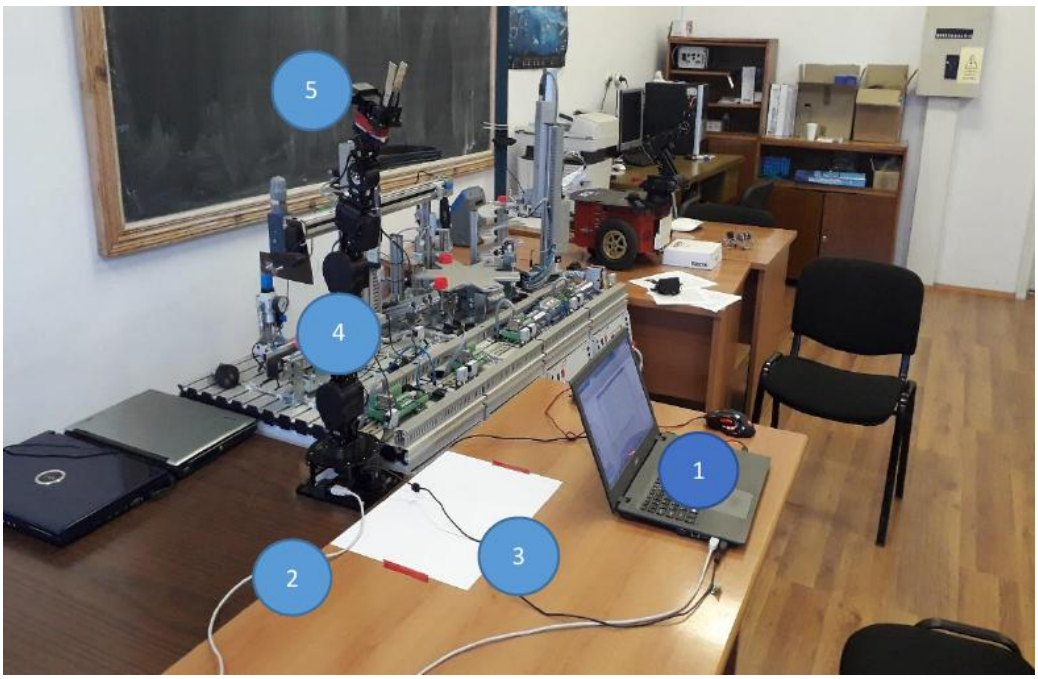

Fig. 2.15.1 Schema fizică a ansamblului: 1-PC, 2-Cablu USB pentru conectare PC-braţ robotic, 3-Cablu USB pentru conectare PC-cameră video, 4-Braţ robotic Cyton Gamma 1500, 5-Cameră video Logitech HD Pro Webcam C920.

O primă etapa este plasarea end-effector-ului braţului robotic la o distanţă de 6cm deasupra obiectelor. După această etapă intră în funcţiune algoritmul pentru prelucrarea imaginii şi de apelare a programelor de control ale braţului robotic. Programul MATLAB este responsabil pentru următoarele funcţii: apelarea programelor pentru comanda braţului robotic, captura imaginii, identificarea obiectelor din imagine, sortarea ariilor obiectelor descrescător şi calculul coordonatelor obiectelor.

După ce braţul robotic a ajuns în poziţia de scanare, se efectuează captura şi prelucrarea imaginii conform schemei prezentate în figura 2.15.2.

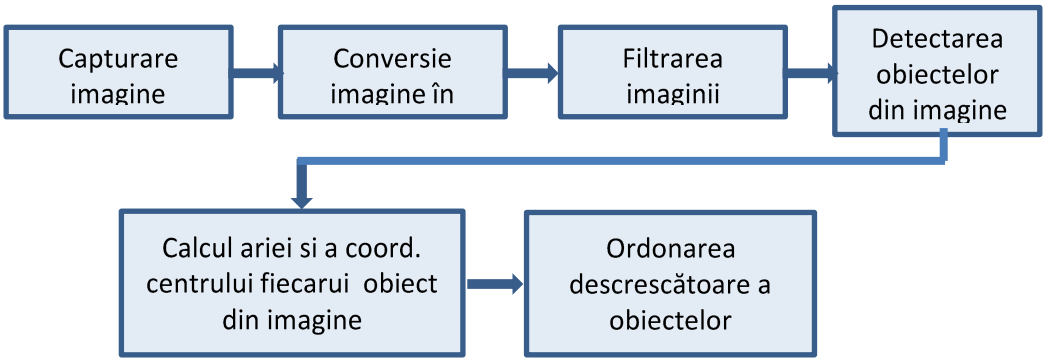

Fig. 2.15.2 Schema algoritmului pentru procesarea imaginii

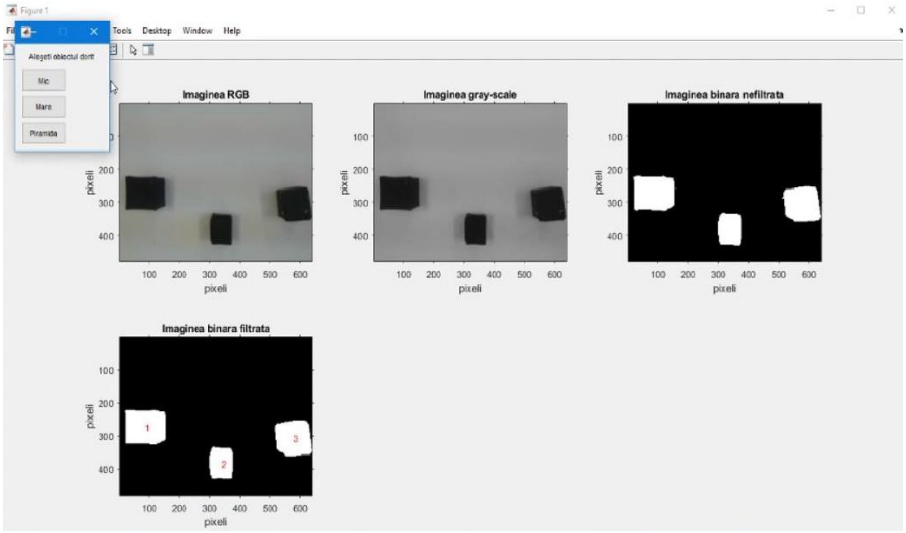

Fig. 2.15.3 Interfaţa grafică realizată în Matlab

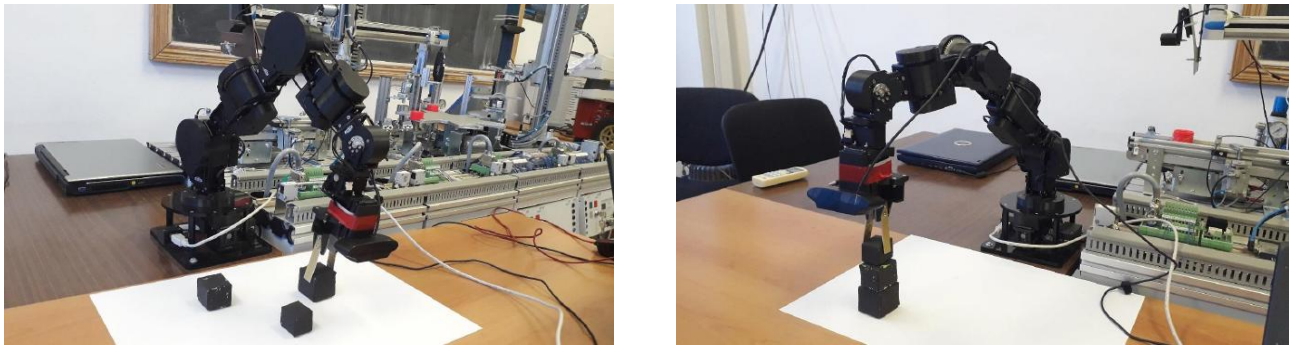

Fig. 2.15.4 Faza iniţială şi cea finală în cazul aranjării obiectelor într-o stivă în ordinea mărimii lor, de la cel mai mare la cel mai mic.

Un al doilea algoritm realizat permite manipularea cutiilor de medicamente (ce au ataşate diverse coduri QR) de către un braţ robotic.

Obiectivele algoritmului au fost i) scanarea unor cutii cu medicamente pe care erau ataşate/lipite coduri QR cu diferite semificaţii, ii) calcularea exacta a poziţiei cutiei de medicamente, iii) trimiterea poziţiei lor către manipulatorul robotic şi iv) manipularea cutiei aleasă de către braţul robotic.

Codul QR (abreviat de la "cod cu răspuns rapid") este un tip de cod de bare matriceală proiectat pentru prima dată in 1994 pentru industria auto din Japonia. Un cod QR constă din pătrate negre aranjate într-o reţea pătrată pe un fundal alb, care poate fi citit de un dispozitiv de imagine, cum ar fi o cameră foto, şi procesată folosind corecţia de eroare Reed-Solomon până când imaginea poate fi interpretată corespunzător. Datele necesare sunt apoi extrase din modelele prezente în componentele orizontale şi verticale ale imaginii.

După detectarea şi decodarea codului QR introdus/cerut de utilizator (a se vedea fig. 2.15.5) se determină poziţia acestuia în imagine (în pixeli). Această poziţie este convertită în centrimetri (a se vedea fig. 2.15.6) şi trimisă către braţul robotic pentru manipulare.

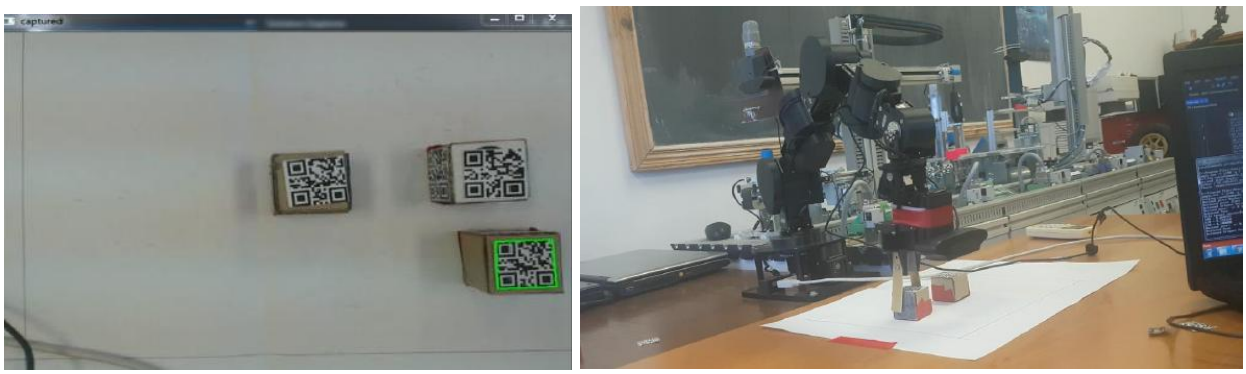

Fig. 2.15.5 Scanarea diverselor coduri QR şi găsirea/detectarea codului cerut de utilizator.

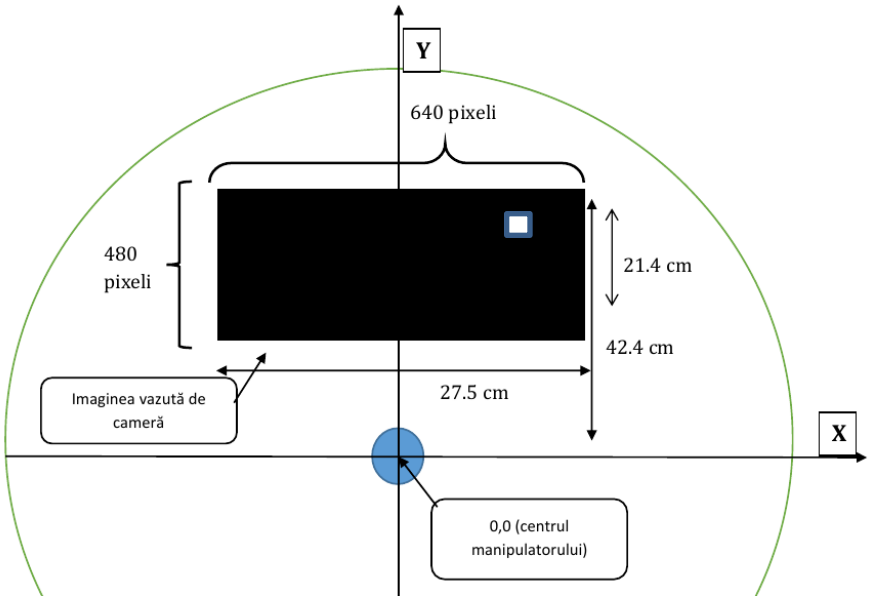

Fig. 2.15.6 Schema de calcul a coordonatelor reale (din pixeli in cm)

Schema algoritmului propus pentru manipularea cutiilor de medicamente (ce au atasate un cod QR) este descrisă în figura 2.15.7.

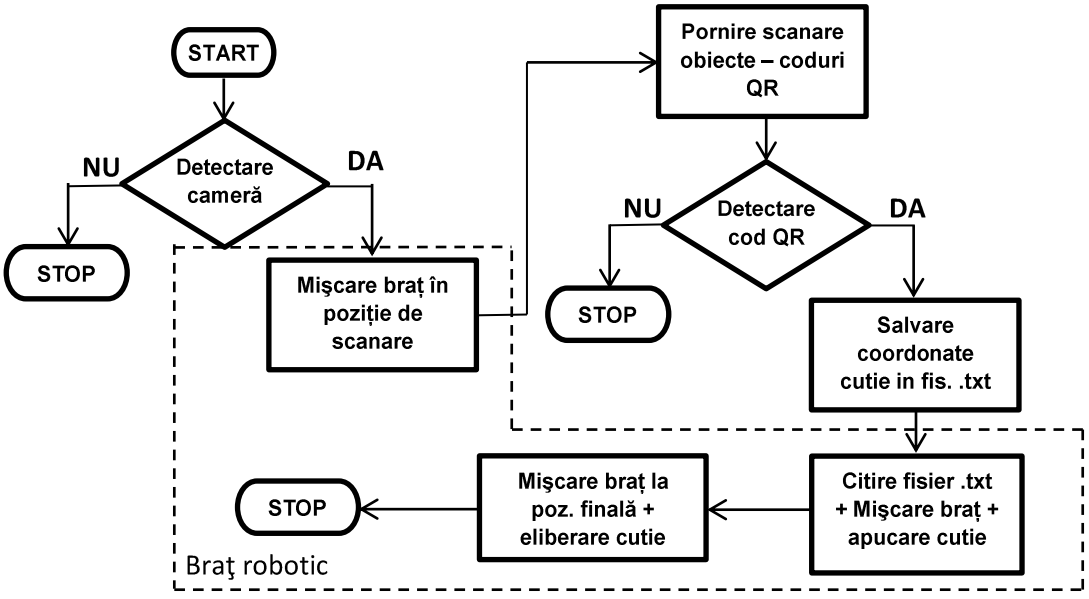

Fig. 2.15.7 Schema algoritm manipulare cutie coduri QR

CONCLUZII

Raportul ştiinţific pune în evidenţă soluţiile pe care echipa de lucru a Proiectului 5 le oferă pentru cerinţele Etapei 2. In Raportul ştiinţific detaliat încărcat pe platforma proiectului P5 (http://www.cidsacteh.ugal.ro), se pot vizualiza soluţiile şi rezultatele cercetări aferente **Etapei 2. "*Proiectarea si integrarea structurilor de conducere inteligenta si distribuita a sistemelor autonome in servicii de deservire medico-sociala si linii de fabricatie de laborator*".**

REZULTATE ETAPA 2

S-au obţinut următoarele rezultate:

- Structură de conducere inteligentă si structură de navigare pentru SAC-SI, "Cirrus Power Wheelchair" integrat în tehnologia de asistare a persoanelor cu dizabilităţi neuro-motorii severe;
- Structură de evitare a obstacolelor (bazată pe senzori de tip laser şi video) ce deserveste scaunul cu rotile de tip "Cirrus Power Wheelchair" (SAC-SI) integrat în tehnologia de asistare a persoanelor cu dizabilităţi neuro-motorii severe;
- Structură de conducere avansată bazată pe senzori performanţi şi sistem servoing vizual mobil a manipulatorului Cyton 1500 care echipează SAC-SI;

DISEMINARE

Articole (ISI Proceedings sau BDI)

1. R. Solea, A. Margarit, D. Cernega, A. Serbencu, "Head Movement Control of Powered Wheelchair", IEEE - 23rd International Conference on System Theory, Control and Computing (ICSTCC), 9-11 Oct. 2019, Sinaia, Romania, DOI: 10.1109/ICSTCC.2019.8885844