

# SISTEM PENTRU NAVIGAREA AUTONOMĂ A ROBOȚILOR MOBILI ÎN BAZA IMAGINILOR GOOGLE MAPS

V. ABABII, V. SUDACEVSCHI, M. PODUBNÎL, I. MOROȘAN, C. ABABII

Universitatea Tehnică a Moldovei

**Abstract:** În lucrarea de față sunt prezentate rezultatele proiectării și implementării unui sistem pentru navigarea autonomă a roboților mobili în baza imaginilor Google Maps. În procesul de proiectare au fost obținute schema de structură și algoritmul de calcul a traiectoriei de deplasare a robotului mobil. Rezultatele proiectării au fost implementate pentru un robot mobil cu trei roți în baza plăcii de dezvoltare Arduino2560 și driverul L298N.

**Cuvinte cheie:** Robot Mobil, navigare autonomă, imagini Google Maps, puncte critice, interpolare, transformări spațiale, Arduino2560, PWM.

## 1. Introducere

Sistemul Google Maps oferă o gamă largă de servicii care permite de a determina amplasarea obiectelor pe suprafața terestră [1]. Aceste servicii sunt prezentate prin imagini sau hărți digitale. Localizarea unui obiect pe hartă sau imagine poate fi efectuată prin metode de descompunere parametrică care include transformări și calcule geometrice, și spațiale [2].

Deplasarea unui robot mobil pe suprafața terestră implică diferite metode de localizare și identificare a mediului înconjurător în scopul excluderii coliziunilor cu obiectele din spațiu și identificarea unui traseu optim de deplasare. În acest scop pe larg sunt utilizate metode de ghidare bazate pe senzori GPS, senzori video, senzori de distanță (laser, ultrasunet, etc.) sau contact, care asigură o calitate satisfăcătoare în procesul de planificare a traseului de deplasare și executarea acestuia [3].

## 2. Sinteza schemei de structură

Schema de structură a Robotului Mobil (RM) este prezentată în Figura 1, unde: **PRM** - platforma Robotului Mobil; **Arduino2560** - placa de dezvoltare în baza de MCU ATmega2560 [4]; **MDCS** și **MDCD** – motoare curent continuu pentru deplasarea RM; **L298N** – driver pentru comanda motoarelor MDCS și MDCD [5]; **RS** și **RD** – două roți pentru deplasarea RM; **RC** – roată de tip castor cu rotație liberă.

**Modul de funcționare.** Mobilitatea RM este asigurată de motoarele MDCS și MDCD. În dependență de viteza de rotație a acestora ( $V_{MDCS}$  și  $V_{MDCD}$ ) se determină și direcția de deplasare a RM. Dacă  $V_{MDCS} = V_{MDCD}$  - RM se deplasează direct înainte, dacă  $V_{MDCS} > V_{MDCD}$  - are loc deplasarea spre dreapta, și dacă  $V_{MDCS} < V_{MDCD}$  - are loc deplasarea spre stânga. Algoritmul de comandă este executat de placa de dezvoltare **Arduino2560** care generează semnale PWM ( $PWM_{MDCS}$  și  $PWM_{MDCD}$ ) pentru ghidarea robotului în spațiul de deplasare. Semnalele generate de **Arduino2560** sunt amplificate de driverul **L298N** și aplicate la bornele motoarelor MDCS și MDCD.

## 3. Algoritmul de calcul a traiectoriei de deplasare

În Figura 2 este prezentat un exemplu de imagine Google Maps, care include: **OXY** – sistemul de coordonate; **A** – punctul de început al deplasării; **B** – punctul de sfârșit al deplasării; **AB** – vectorul direcției de deplasare din punctul **A** în punctul **B**; **Obs** – obstacole în spațiul de navigare; **C1 – C10** – puncte critice

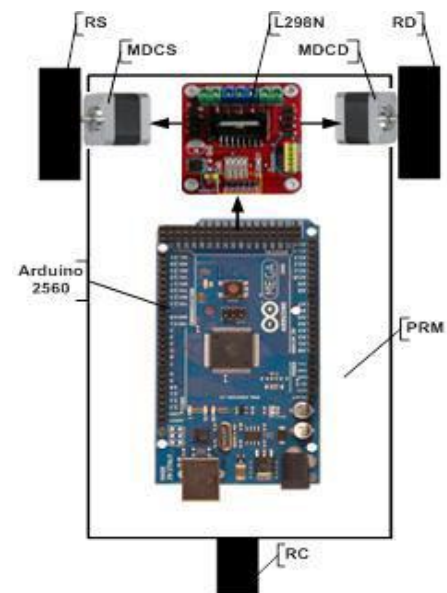


Fig. 1. Schema de structură a Robotului Mobil.

identificate în spațiul de navigare;  $AB^*$  - traiectoria de deplasare a RM obținută în rezultatul interpolării punctelor critice  $C1 - C10$ .

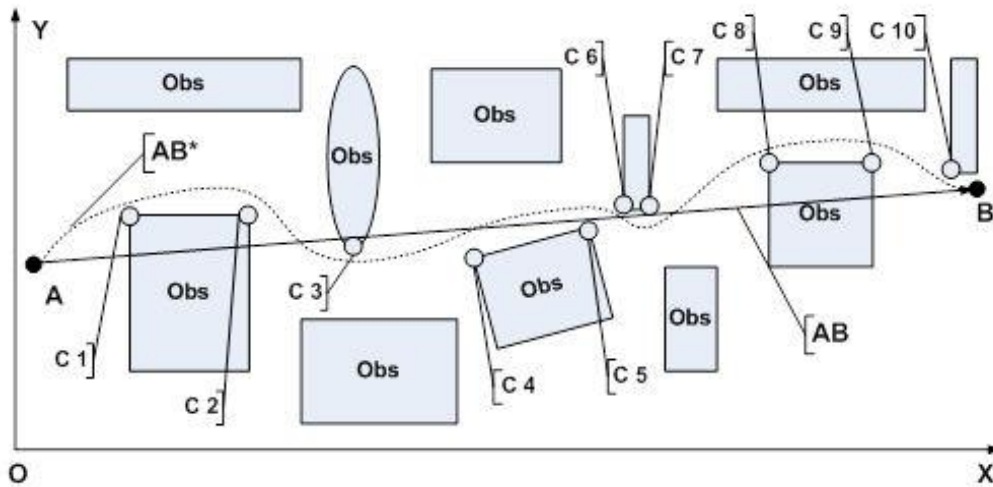


Fig. 2. Exemplu de imagine Google Maps pentru calculul traiectoriei de deplasare a RM.

În Figura 3 este prezentat rezultatul transformărilor geometrice efectuate asupra schemei din Figura 2 pentru a fi adaptat la algoritmul de ghidare a RM în spațiul de navigare, unde:  $O'X'Y'$  - sistemul de coordonate după transformările geometrice;  $AB'$  - vectorul direcției de deplasare din punctul A în punctul B' care corespunde cu axa  $O'X'$  a sistemului de coordonate;  $C'1 - C'10$  - puncte critice după transformările geometrice;  $AB'^*$  - noua traiectorie de deplasare a RM.

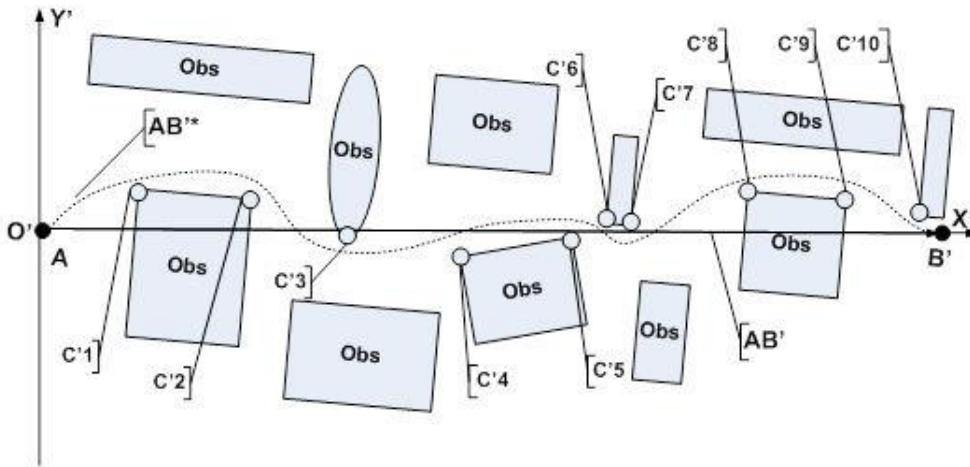


Fig. 3. Rezultatul transformărilor geometrice efectuate asupra exemplului din Figura 2.

**Algoritmul de calcul a traiectoriei de deplasare a RM** include următorii pași (Figura 4):

1. Începutul algoritmului;
2. Achiziția imaginii Google Maps care include obiectele din spațiul de navigare a RM;
3. Filtrarea și conturarea obstacolelor (*Obs*) din spațiul de navigare (Figura 2);
4. Identificarea vectorului de deplasare directă  $AB$  din punctul  $A$  în punctul  $B$ ;
5. Identificarea punctelor critice ( $C1 - C10$ );
6. Calculul traiectoriei de deplasare a RM prin metoda interpolării în baza punctelor critice  $C1 - C10$ ;
7. Efectuarea transformărilor geometrice care includ: deplasarea originii sistemului de coordonate  $O'X'Y'$  în punctul  $A$ ; rotirea spațiului de navigare până la coincidența direcției de deplasare  $AB'$  a RM cu axa  $O'X'$  (Figura 3);
8. Calculul vitezei de rotație  $V_{MDCS}$  și  $V_{MDCD}$  a motoarelor de deplasare a RM în baza formulei:

$$\left\{ \begin{array}{l} \left( (V_{MDCS} = M * 1 * T) \& (V_{MDCD} = M * 1 * T) \right) \left| \frac{d^2 y}{dx^2} = 0; \right. \\ \left( (V_{MDCS} = M * \cos(\alpha) * T) \& (V_{MDCD} = M * 1 * T) \right) \left| \frac{d^2 y}{dx^2} > 0; \right. \\ \left( (V_{MDCS} = M * 1 * T) \& (V_{MDCD} = M * \cos(\alpha) * T) \right) \left| \frac{d^2 y}{dx^2} < 0. \right. \end{array} \right. \quad (1)$$

Unde în formula (1) avem:  $M$  - coeficientul scării cartografice [6];  $T$  - valoarea maximală a timpului pentru modularea PWM pentru comanda motoarelor  $MDCS$  și  $MDCD$ ;  $\frac{d^2 y}{dx^2}$  - a doua derivată a funcției de interpolare a traiectoriei de deplasare a RM (Dacă  $\frac{d^2 y}{dx^2} = 0$  - are loc deplasarea RM drept înainte. Dacă  $\frac{d^2 y}{dx^2} > 0$  - are loc cotirea RM spre dreapta. Dacă  $\frac{d^2 y}{dx^2} < 0$  - are loc cotirea RM spre stânga);  $\alpha$  - unghiul de variație a direcției de deplasare a RM în raport cu valoarea precedentă ( $\alpha = \arctg\left(\frac{d^2 y}{dx^2}\right)$ ).

9. Aplicarea semnalelor de comandă la bornele motoarelor  $MDCS$  și  $MDCD$ ;
10. Dacă nu este atinsă coordonata punctului  $B$ , salt la p.8. Dacă este atinsă coordonata punctului  $B$ , salt la p.11;
11. Sfârșitul algoritmului.

Algoritmul de calcul a traiectoriei de deplasare a robotului mobil a fost implementat pentru placa de dezvoltare **Arduino2560** în mediul de programare **MicroC PRO for AVR 4.60** (Advanced C Compiler for AVR MCU) [7].

#### 4. Concluzii

În rezultatul proiectării și implementării sistemului pentru navigarea autonomă a roboților mobili în baza imaginilor Google Maps au fost elaborate schema de structură și algoritmul de calcul a traiectoriei de deplasare a robotului mobil. Au fost obținute modelele matematice pentru calculul semnalelor de comandă cu motoarele de deplasare pentru a atinge o convergență maximală a traiectoriei calculate și a traiectoriei reale de deplasare a robotului mobil.

#### Mențiuni

Proiectările și cercetările au fost efectuate cu suportul tehnic oferit de Centrul de Creativitate Tehnică „Hard & Soft” din cadrul Facultății Calculatoare, Informatică și Microelectronică al Universității Tehnice a Moldovei.

#### Bibliografie

24. <https://www.google.ro/maps/> (Accesat 12.10.2015).
25. Boja, N. Geometrie analitică și diferențială. Editura Politehnica, 2005.
26. Țărulescu R. Contribuții privind optimizarea configurației senzorilor utilizați la roboții mobili. Teză de doctorat. Universitatea Transilvania din Brașov. Brașov, 2014, p. 71.
27. <https://www.arduino.cc/en/Main/ArduinoBoardMega2560> (Accesat 23.10.2015).
28. <http://www.st.com/web/en/resource/technical/document/> (Accesat 25.10.2015).
29. [https://ro.wikipedia.org/wiki/Scara\\_\(cartografie\)](https://ro.wikipedia.org/wiki/Scara_(cartografie)) (Accesat 29.10.2015).
30. <http://www.mikroe.com/> (Accesat 29.10.2015).

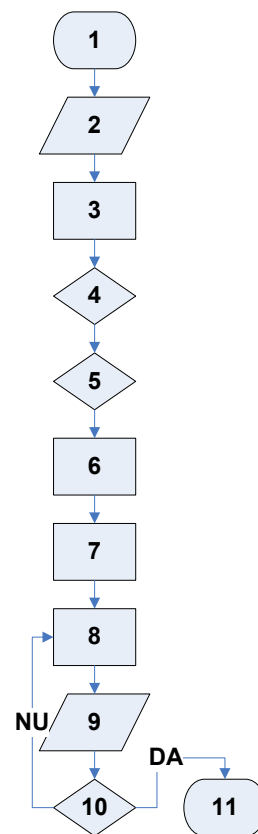


Fig. 4. Algoritmul de calcul a traiectoriei de deplasare a RM.