

PROIECTAREA ȘI REALIZAREA APLICAȚIEI SOFTWARE ÎN VEDEREA DETERMINĂRII PRECIZIEI DE POZIȚIONARE

DRĂGHICIU Cristian

Conducători științifici: Conf. dr. ing. **George ENCIU**, Asist. univ. dr. ing. **Adrian POPESCU**

REZUMAT: AGV-ul cu sistem de transfer are în componență un subansamblu pentru urmărirea liniei (line follower), ce se deplasează pe conform unui traseu (cale de rulare) reprezentat de o linie neagră pe un fundal alb. Linia se trasează cu o bandă izolieră pe platforma de lucru, deasupra căreia va fi lipit un colant alb, iar pentru o aderență mai bună colantul va prezenta striățiuni dispuse perpendicular unele față de altele.

CUVINTE CHEIE: AGV, urmaritor de linie, Arduino,

1. INTRODUCERE

În vederea obținerii preciziei de poziționare a AGV-ului cu sistem de transfer s-au încercat mai mulți algoritmi, printre care și algoritmul PID (eng. Proportional Integral Derivative). În urma aplicării și rulării acestui algoritm, AGV-ul urmărește calea de ghidare cu o precizie ridicată.

2. STADIUL ACTUAL

Citirea liniei după care se deplasează AGV-ul se face cu ajutorul sistemului senzorial QTR8A, fiind format din opt senzori analogici QRE1113, alcătuiți din opt led-uri cu lumina infraroșie și opt fototranzistori, sensibili la lumina infraroșie.

Specializarea Logistică Industrială, Facultatea IMST;

Email: draghiciucristian@imst.pub.ro;

1.1 Citirea propriu-zisă

Semnalele provenite de la senzori sunt trimise spre procesare Unității Centrale reprezentată de Platforma de dezvoltare Arduino Uno ce conține microcontroller-ul ATmega328 de la ATMEL. Platforma de dezvoltare Arduino Uno reprezintă sistemul de comandă al AGV-ului și conține 14 pini digitali de intrare-ieșire și 6 pini analogici. Pentru AGV s-au folosit următorii pini: analogici: A0, A1, A2, A3, A4, A5 conectați fiecare la pinul de ieșire al fiecărui senzor de linie, de la cei din centru spre extremități și digitali: D4 și D7 folosiți pentru primul și ultimul senzor; D3, D5, D6 și D11 folosiți pentru controlul motoarelor electrice de curent continuu, iar D8 este folosit pentru interceptarea unui semnal de pornire provenit de la altă platformă Arduino Uno, impuls oferit în urma efectuării unei inițializări.

1.2 Comanda motoarelor

Semnalele procesate sunt apoi transmise către shield-ul L298 (vezi figura 1) ce comandă,

conform principiului unei punți H, cele două motoare de curent continuu ale AGV-ului.

1.2.1 Ce este o punte H?

O punte H (eng. H Bridge) este un circuit electronic ce permite aplicarea unei tensiuni pe o sarcină în orice sens. Aceste circuite sunt adesea folosite în robotică și alte aplicații pentru a permite motoarelor de curent continuu să ruleze înainte și înapoi. Punțile H sunt disponibile ca circuite integrate sau pot fi construite din componente discrete, tranzistoare bipolare sau MOS.



Fig. 1. Shield L298

Puntea H are numele derivat de la modul obișnuit de desenare a circuitului (vezi figura 2). Aceasta este singura cale de tip solid state de a comanda motorul în ambele direcții.

Proiectarea și realizarea aplicației software în vederea derterminării preciziei de poziționare

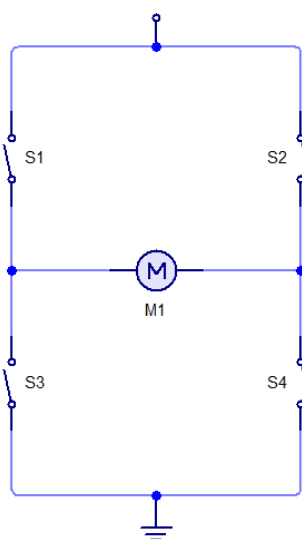


Fig. 2. Schema de principiu a punții H

Mod de funcționare:

Atunci când întrerupătoarele S1 și S4 sunt închise și S2 și S3 sunt deschise o tensiune pozitivă va fi aplicată la nivelul motorului. (vezi figura 3.1)

Prin deschiderea întrerupătoarelor S1 și S4 și închiderea întrerupătoarelor S2 și S3, această tensiune este inversată, astfel să permită funcționarea inversă a motorului. (vezi figura 3.2)

Folosind nomenclatura de mai sus, întrerupătoarele S1 și S2 nu trebuie să fie închise în același timp, deoarece acest lucru ar provoca un scurt-circuit la sursa de tensiune (V_{in}). Același lucru se aplică și întrerupătoarelor S3 și S4.

În practică întrerupătoarele S1,S2,S3,S4 sunt tranzistoare bipolare sau MOS-FET.

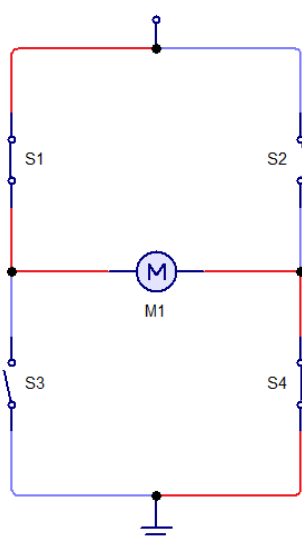


Fig. 3.1. Alimentarea motorului electric intr-un sens

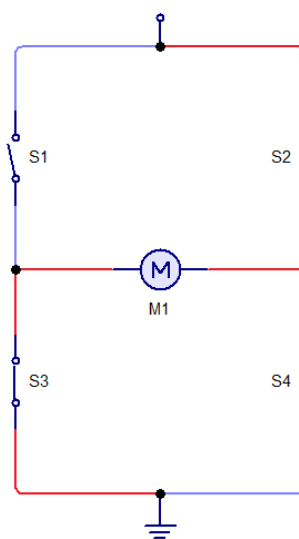


Fig. 3.2. Alimentarea motorului electric sens opus

1.2.2 Circuitul integrat L298

Circuitul integrat L298 (vezi figura 4) este un driver de nivel mediu din punct de vedere al puterii conduse. Poate controla ușor și independent motoare care necesita cel mult 2 Amperi, în ambele direcții. Este un driver de motoare dual și bidirecțional (punte H dublă).

Specificatii:

- Tensiune de operare - maxim 46V
- Tensiune de alimentare - 4.5-7V
- Curent maxim - 4A
- Tensiune de saturație scăzută
- Protecție la supraîncălzire
- Intrarea logică „0” - maxim 1.5V

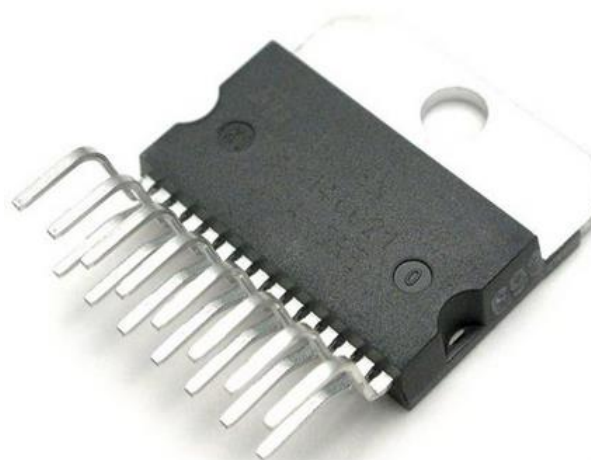


Fig. 4. Circuitul integrat L298

3. ALGORITMUL PID

Precizia de poziționare a AGV-ului se realizează cu ajutorul algoritmului PID.

Regulatele PID sunt unele din cele mai complexe regulate industriale, asigurând performanțe de reglare superioară.

Caracteristici:

- funcționează atât în regim staționar, cât și în regim tranzitoriu
- au trei parametri ajustabili K_P , K_I , K_D
- asigură posibilități mult mai largi în asigurarea legilor de reglare
- explică performanțele superioare ale sistemelor de reglare automată

Algoritmul PID funcționează după următoarea ecuație:

$$c(t) = K_P \left[a(t) + \frac{1}{T_I} \int_0^t a(t) \cdot dt + T_D \cdot \frac{da}{dt} \right]$$

Unde termenii au semnificațiile:

- $c(t)$ - semnal de comandă
- $a(t)$ - semnal de abatere
- K_P - factor de proportionalitate
- T_I - constanta de integrare
- T_D - constanta derivativă

Ultimii trei termeni se numesc parametri de acordare (tuning) sau de reglare.

Ecuația anterioară se mai poate scrie astfel:

$$c(t) = K_P \cdot a(t) + K_I \int_0^t a(t) \cdot dt + K_D \cdot \frac{da}{dt}$$

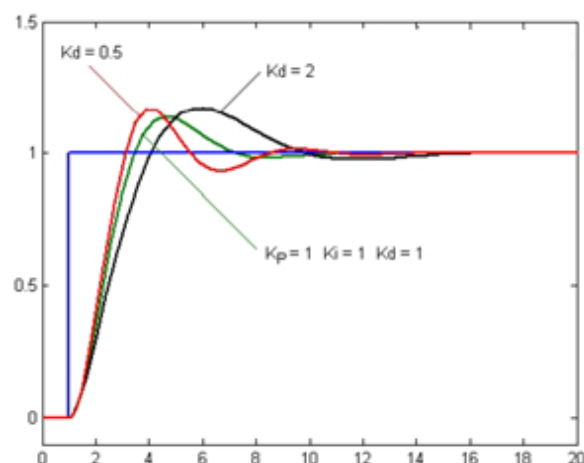
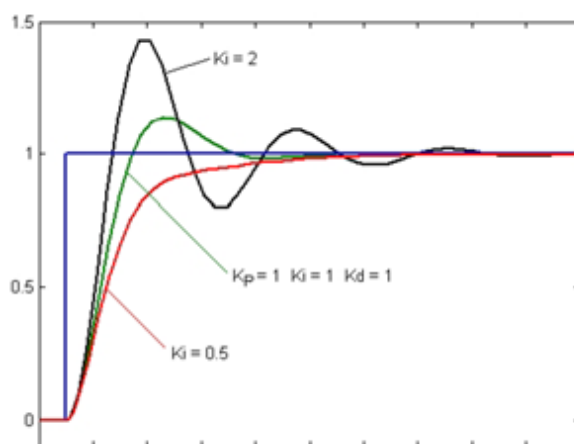
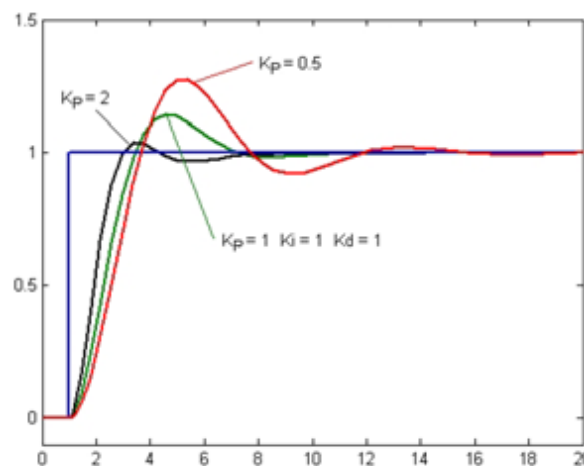
Unde:

$$\begin{cases} K_I = \frac{K_P}{T_I} \\ K_D = K_P \cdot T_D \end{cases}$$

1.2.3 Modificarea răspunsului în urma modificării parametrilor K_P , K_I și K_D

Legenda:

- Semnal de referință
- Semnal de comandă pentru diferite valori pentru parametrul K_P , K_D și K_I



Proiectarea și realizarea aplicației software în vederea derterminării preciziei de poziționare

4. PLATFORMA ARDUINO UNO

Arduino UNO (vezi figura 5) este o platformă de procesare open-source, bazată pe software și hardware flexibil și simplu de folosit. Constă într-o platformă de mici dimensiuni (6.8 cm / 5.3 cm – în cea mai des întâlnită variantă) construită în jurul unui procesor de semnal și este capabilă de a prelua date din mediul înconjurător printr-o serie de senzori și de a efectua acțiuni asupra mediului prin intermediul luminilor, motoarelor, servomotoarelor, și alte tipuri de dispozitive mecanice. Procesorul este capabil să ruleze cod scris într-un limbaj de programare care este foarte similar cu limbajul C++.

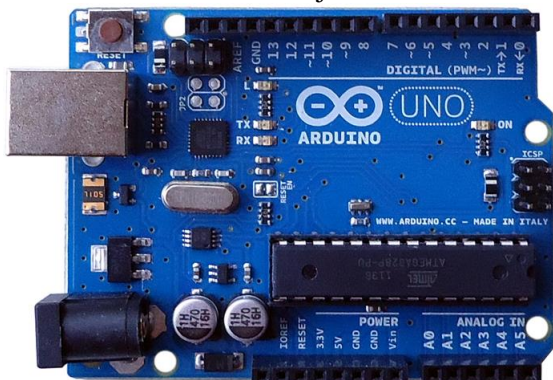


Fig. 5. Platforma Arduino Uno

5. APLICAȚIA PROPRIU-ZISĂ

1.3 Limbajul de programare utilizat

Aplicația software de comandă a AGV-ului, din motive de necesitate, este realizată în limbajul AVR Studio. (vezi figura 6)

1.4 Implementarea algoritmului PID

```
int i=1;
int eroaretotala=0;
int proportional = obtine_eroarep();
int integral = obtine_eroarei();
int derivativ = obtine_eroared();
eroaretotala = proportional + integral + derivativ;
if(eroaretotala > viteza)
    eroaretotala = viteza;
else if(eroaretotala < -viteza)
    eroaretotala = -viteza;
else if(((PINC & 0x04) != 0) && ((PINC & 0x08) != 0)){
    M1_fata(viteza);
    M2_fata(viteza);}
else if(eroaretotala == viteza){
    M1_sparte(viteza/2);
    M2_fata(viteza);}
else if(eroaretotala == -viteza) {
    M1_fata(viteza);
    M2_sparte(viteza/2);}
else if(eroaretotala>0 && eroaretotala < viteza){
    M1_fata(viteza - eroaretotala);
    M2_fata(viteza);}
else if(eroaretotala<0 && eroaretotala > -viteza){
    M1_fata(viteza);
    M2_fata(viteza + eroaretotala); }
else{
    M1_fata(viteza);
    M2_fata(viteza);
}
```

```
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>

//Comunicarea între platforme
#define STR PORTB1 //Comanda pornire transfer
#define STP PORTB2 //Confirmare transfer
```

Fig. 6.1. Secvență din aplicația software

6. CONCLUZII

În concluzie, pentru urmărirea fidelă a liniei, fără abateri care pot determina pierderea controlului, este nevoie de un algoritm complex și stabil din punct de vedere al funcționalității.

În cazul în care AGV-ul se deplasează prea mult într-o direcție, fapt în urma căruia pierde linia, fără o aplicație software care să corecteze eroarea și să-l readucă deasupra liniei, AGV-ul va vira chiar în direcția în care a ieșit de pe linie.

Regulatorul PID de bază compară o singură intrare cu o valoare de referință, și calculează o valoare de ieșire unică pentru a controla sistemul, iar algoritmul presupune că răspunsul va fi liniar, și nu se modifică în timp. Există totuși, anumite tehnici de aplicare al acelorași algoritmi PID la sisteme care nu îndeplinesc neapărat aceste criterii.

7. BIBLIOGRAFIE

- [1]. Massimo Banzi și Michael Shiloh (2014), *Getting Started With Arduino 3rd Edition*, Maker Media, Malmö, ISBN 13: 978-1-4493-0987-9
- [2]. T. Olsson, D. Gaetano, J. Odhner, și S. Wiklund(2011), *Open Softwear 2nd Edition*, Blushing Boy Publishing, Malmö, ISBN: 978-91-97-95540-9
- [3]. *Part 1:: Line following robot :: code guideline arduino using PID*, disponibil: <https://syclops.wordpress.com/2011/08/06/line-following-robot-code-guideline-using-pid/>
Accesat la data: 28.10.2014
- [4]. *Part 2:: line following robot :: code guideline arduino using PID*, disponibil: <https://syclops.wordpress.com/2011/08/07/part-2-line-following-robot-code-guideline-arduino-using-pid-2/>
Accesat la data: 28.10.2014