

Robot Teknolojileri Dersi Proje Ödevi

Konu : Kontrol Algoritması Geliştirme

Amaç : Çizgi izleyen robotlar için simülatör üzerinde koşturulacak kontrol algoritması geliştirerek, farklı pistlerde robot kontrolü sağlamak.

Genel Bilgi : Robotlar, kontrol algoritmaları ile hareket etmektedir. Amaç robotu en kısa sürede bitiş noktasına iletmektir. Robot ilerlerken zemindeki çizgiyi takip eder ve çizgi pixellerini sayar. Döngü süresi, yani algoritmanın çağrılma sayısı, düşük olan robot yarışı kazanır.

Simülatörün robotları farklı karakterdedir. Robotların karakterlerini belirleyen unsurlar şunlardır.

- 1) Teker çapı
- 2) Robotun uzunluğu
- 3) Tekerlekler arası mesafe
- 4) Redüktör oranı
- 5) Robotun kütlesi
- 6) Sürtünme değeri
- 7) Motorların karakteristiği
- 8) Algoritma

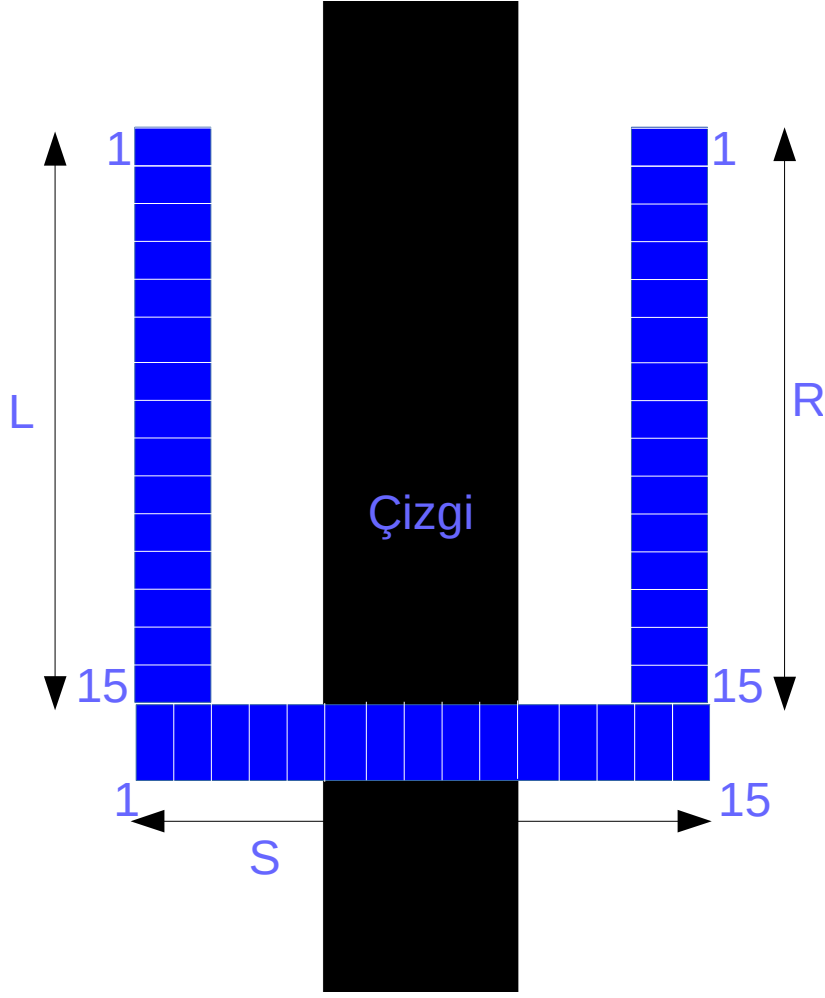
Robot, sol teker ve sağ teker için bağımsız kontrol edilen iki motor sayesinde hareket eder. Tekerlerin dönme sayısı eşit ise robot düz ilerler. Tekerlerden birisi daha fazla dönerse robot diğer teker yönüne doğru dairesel kavis çizer. Tekerlerin dönme sayısı ve dönme yönü, motorlara uygulanan gerilim ile kontrol edilir. Motor gerilimleri 8 bit PWM ile değiştirilir. Motora uygulanan voltaj değiştirildiğinde motorun devir sayısı, gerçek dünyada olduğu gibi yavaş etkilenir.

Robotun sensörleri tüm renkleri algırlar ancak algoritmanıza sadece rengin yol çizgisi rengi olup olmadığını belirtir. Robot sensörü, algoritmanızın değişkenlerine, yol çizgisi renginde bir pixel ile karşılaşırsa 1, diğer renklerde bir nokta ile karşılaşırsa 0 bilgisini bildirir.

Robotun 3 sensör bloğu vardır.

- | | |
|--------------------------------------|-------------|
| 1. Grup : robotun önünde yatay | (S grubu) |
| 2. Grup : robotun önünde solda dikey | (L grubu) |
| 3. Grup : robotun önünde sağda dikey | (R grubu) |

Her grup aşağıdaki şekilde yerleştirilmiş 15 sensörden oluşur.



Sensorler yukarıdaki gösterimde olduğu gibi U şeklinde dizilmiştir ve robot gövdesi bu sensör bloğunun gerisinde kalır. Çizgi genişliği 5 pixeldir ve S grubu 15 pixel genişliğindedir. Her bir sensör 1 pixelden etkilenir. L ve R grubu 15 adet 1 pixel genişliğindeki sensörlerin alt alta dizilmesi ile oluşturulmuştur.

Zemin üzerindeki çizgiyi tam ortalamış olan robotun sadece S grubu sensörleri çizgiyi algılar. Ancak robot çizgiden ayrılırsa yada çizgi yönünde çok keskin dönüş varsa L yada R grubu sensörler bunu algılar.

L ve R grubu, çizginin keskin yön değişikliklerini algılamak için eklenmiştir. Gerçek çizgi izleyen robotlardada genellikle sadece S grubu sensörler bulunur.

Algoritma yapısı

Sensör bilgilerine göre motor voltaj değerlerini belirleyen yazılımı siz yazacak ve dll fonksiyonu haline getireceksiniz. Bu dll'in adı MyAlgo.dll olmalıdır.

MyAlgo.dll içinde 7 adet algoritma fonksiyonu olmalıdır. Bu fonksiyonlar Algo1, Algo2, Algo3 Algo7 olarak isimlendirilmelidir. Bu fonksiyonlardan hangisinin kullanılacağı simülatördeki algoritma seçim menüsünden belirlenir. Sadece tek bir algoritma yazacak olmanıza rağmen diğer fonksiyonları da oluşturmak zorundasınız. Sizden istenen yalnızca bir adet algoritma yazmanızdır. Diğer 6 algoritma, hiçbir kontrol yapmayacak şekilde sabit hızda düz gidecek şekilde çalışmalıdır.

Yazacağınız algoritmalar S, L, R, Init, VL, ve VR olmak üzere toplam 5 adet parametre alacaktır. S, L, R sensör sinyallerini barındıran değişkenlerdir ve içerikleri robotun pistteki durumuna göre simülatör tarafından oluşturulur. Yalnızca S kullanarak da algoritmanızı geliştirebilirsiniz. L ve R kullanmak isteğe bağlıdır. Init sadece yarış başladığında bir kez 1 olur. VL ve VR sağ ve sol motorlara ait voltaj değişkenleridir. VL ve VR değerleri algoritmanız tarafından oluşturulur. Eğer algoritma VL ve VR değişkenlerine değer yüklemmezse bir önceki VL ve VR değerleri geçerliliğini korur.

Aşağıda 2 adet örnek algoritma verilmiştir. Bu örnekler yalnızca temel yapıyı göstermek amaçlı olup algoritma yapısını **göstermemektedir**. Yazacağınız algoritmada yapmanız gereken, S, L, ve R sensörlerinden alınan veriyle **PID** kontrol sistemi tasarlayarak kontrol algoritmasını işletmeniz ve robotun çizgiyi takip ederek en kısa sürede hatasız olarak bitiş çizgisine varmasını sağlamaktır. Algoritmanızı istediğiniz herhangi bir programlama dilinde yazarak dll dosyasını oluşturmanız ve simülatör ile denemeniz gerekmektedir. **TEKRAR : Aşağıdaki örnek kodlar yalnızca yapıyı göstermek içindir. Yazacağınız algoritma bu mantıkla çalışmayacaktır. PID kontrol sistemi tasarlamanız gerekmektedir.**

Delphi ile yazılmış örnek algoritma:

```
library MyAlgo;

uses
  SysUtils,
  Classes;

{$R *.res}

const
  B000001111100000 = $03E0;

  B000011111000000 = $07C0;
  B000111110000000 = $0F80;
  B001111100000000 = $1F00;
  B011111000000000 = $3E00;
  B111110000000000 = $7C00;
  B111000000000000 = $F800;
  B110000000000000 = $E000;
  B100000000000000 = $6000;
  B100000000000000 = $4000;

  B000000111110000 = $01F0;
  B000000011111000 = $00F8;
  B000000001111100 = $007C;
  B000000000111110 = $003E;
  B000000000011111 = $001F;
  B000000000001111 = $000F;
  B000000000000111 = $0007;
  B000000000000011 = $0003;
  B000000000000001 = $0001;

  B111111111111111 = $7FFF;
  B011111111111110 = $3FFE;
  B001111111111100 = $1FFC;
  B000111111111000 = $0FF8;
  B000011111110000 = $07F0;

{
  Robotun onunde soldan saga dogru 15 adet sensor dizilidir.
  En soldaki sensor bilgisi 14.Bitde, En sagdaki sensor bilgisi 0. bittedir.
  Sol motora uygulayacaginiz voltajı VL icine yaziniz
  Sag motora uygulayacaginiz voltajı VR icine yaziniz
  Motor voltajı -127 ... + 127 araligindadir.
}

Algol(S,L,R,Init: Integer; var VL,VR: Integer):boolean; stdcall; //***** ALGO 1
var i,n,m:integer;
begin
  if S = B111111111111111 then begin VL:=127; VR:=127;end;
  if S = B011111111111110 then begin VL:=127; VR:=127;end;
  if S = B001111111111100 then begin VL:=127; VR:=127;end;
  if S = B000111111111000 then begin VL:=127; VR:=127;end;
  if S = B000011111110000 then begin VL:=127; VR:=127;end;

  if S = B000001111100000 then begin VR:=127; VL:=127;end; // TAM ORTALAMIS GIDIYORUZ

  if S = B000011111000000 then begin VL:=64; VR:=127;end;
  if S = B000111110000000 then begin VL:=32; VR:=127;end;
  if S = B001111100000000 then begin VL:=16; VR:=127;end;
  if S = B011111000000000 then begin VL:=8; VR:=127;end;
  if S = B111110000000000 then begin VL:=4; VR:=127;end;
  if S = B111100000000000 then begin VL:=2; VR:=127;end;
  if S = B111000000000000 then begin VL:=1; VR:=127;end;
  if S = B110000000000000 then begin VL:=0; VR:=127;end;
  if S = B100000000000000 then begin VL:=0; VR:=127;end;

  if S = B000000111110000 then begin VR:=64; VL:=127;end;
  if S = B000000011111000 then begin VR:=32; VL:=127;end;
  if S = B000000001111100 then begin VR:=16; VL:=127;end;
  if S = B000000000111110 then begin VR:=8; VL:=127;end;
  if S = B000000000011111 then begin VR:=4; VL:=127;end;
  if S = B000000000001111 then begin VR:=2; VL:=127;end;
  if S = B000000000000111 then begin VR:=1; VL:=127;end;
  if S = B000000000000011 then begin VR:=0; VL:=127;end;
  if S = B000000000000001 then begin VR:=0; VL:=127;end;

  n:=0; m:=0;

  for i:=14 downto 7 do
    begin
      if (L and (1 SHL i))<>0 then n:=n+1;
      if (R and (1 SHL i))<>0 then m:=m+1;
    end;

  if n>m then
    begin
```

```

        VL:=0;
        VR:=64;
    end;

    if n<m then
        begin
            VL:=64;
            VR:=0;
        end;

        result:=true;
    end;

var PL,IL:extended;
VL, VR:extended;

function Algo2(S,L,R,Init: Integer; var VL,VR: Integer):boolean; Stdcall; //***** ALGO 1
begin
    result:=Algol(S,L,R,VL,VR,Init);
end;

function Algo3(S,L,R,Init: Integer; var VL,VR: Integer):boolean; Stdcall; //***** ALGO 1
begin
    result:=Algol(S,L,R,VL,VR,Init);
end;

function Algo4(S,L,R,Init: Integer; var VL,VR: Integer):boolean; Stdcall; //***** ALGO 1
begin
    result:=Algol(S,L,R,VL,VR,Init);
end;

function Algo5(S,L,R,Init: Integer; var VL,VR: Integer):boolean; Stdcall; //***** ALGO 1
begin
    result:=Algol(S,L,R,VL,VR,Init);
end;

function Algo6(S,L,R,Init: Integer; var VL,VR: Integer):boolean; Stdcall; //***** ALGO 1
begin
    result:=Algol(S,L,R,VL,VR,Init);
end;

function Algo7(S,L,R,Init: Integer; var VL,VR: Integer):boolean; Stdcall; //***** ALGO 1
begin
    result:=Algol(S,L,R,VL,VR,Init);
end;

exportS  Algol,Algo2,Algo3,Algo4,Algo5,Algo6,Algo7;

begin
end.

```

C++ ile yazılmış örnek algoritma

```
#include "myalgo.h"

#include <cstdlib>

bool Algo1(unsigned short S, unsigned short L, unsigned short R, int Init, int * VL, int * VR)
{
    if (S == 0b1111111111111111) { * VL=50;* VR = 50;}
    if (S == 0b0111111111111110) { * VL=50;* VR = 50;}
    if (S == 0b0011111111111100) { * VL=50;* VR = 50;}
    if (S == 0b0001111111111000) { * VL=50;* VR = 50;}
    if (S == 0b0000111111110000) { * VL=50;* VR = 50;}

    if (S == 0b0000011111100000) { * VL=50;* VR = 50;}

    if (S == 0b0000111110000000) { * VL=64;* VR = 50;}
    if (S == 0b0001111100000000) { * VL=32;* VR = 50;}
    if (S == 0b0011111000000000) { * VL=16;* VR = 50;}
    if (S == 0b0111110000000000) { * VL=8; * VR = 50;}
    if (S == 0b1111100000000000) { * VL=4; * VR = 50;}
    if (S == 0b1111000000000000) { * VL=2; * VR = 50;}
    if (S == 0b1110000000000000) { * VL=1; * VR = 50;}
    if (S == 0b1100000000000000) { * VL=0; * VR = 50;}
    if (S == 0b1000000000000000) { * VL=0; * VR = 50;}

    if (S == 0b0000000111110000) { * VR=64; * VL = 50;}
    if (S == 0b0000000011111000) { * VR=32; * VL = 50;}
    if (S == 0b0000000001111100) { * VR=16; * VL = 50;}
    if (S == 0b0000000000111110) { * VR=8; * VL = 50;}
    if (S == 0b0000000000011111) { * VR=4; * VL = 50;}
    if (S == 0b0000000000001111) { * VR=2; * VL = 50;}
    if (S == 0b0000000000000111) { * VR=1; * VL = 50;}
    if (S == 0b0000000000000011) { * VR=0; * VL = 50;}
    if (S == 0b0000000000000001) { * VR=0; * VL = 50;}

    return true;
}

bool Algo2(unsigned short S, unsigned short L, unsigned short R, int Init, int * VL, int * VR)
{
    // Robot sadece düz gidecek. Sensör kontrolü yok.
    *VL = 5;
    *VR = 5;
}

bool Algo3(unsigned short S, unsigned short L, unsigned short R, int Init, int * VL, int * VR)
{
    return Algo1(S, L, R, Init, VL, VR);
}

bool Algo4(unsigned short S, unsigned short L, unsigned short R, int Init, int * VL, int * VR)
{
    return Algo1(S, L, R, Init, VL, VR);
}

bool Algo5(unsigned short S, unsigned short L, unsigned short R, int Init, int * VL, int * VR)
{
    return Algo1(S, L, R, Init, VL, VR);
}

bool Algo6(unsigned short S, unsigned short L, unsigned short R, int Init, int * VL, int * VR)
{
    return Algo1(S, L, R, Init, VL, VR);
}

bool Algo7(unsigned short S, unsigned short L, unsigned short R, int Init, int * VL, int * VR)
{
    return Algo1(S, L, R, Init, VL, VR);
}
```

```
#ifndef MYALGO_H
#define MYALGO_H

#include <math.h>

extern "C" __declspec(dllexport) bool Algo1(unsigned short S, unsigned short L, unsigned short R, int Init, int * VL, int * VR);
extern "C" __declspec(dllexport) bool Algo2(unsigned short S, unsigned short L, unsigned short R, int Init, int * VL, int * VR);
extern "C" __declspec(dllexport) bool Algo3(unsigned short S, unsigned short L, unsigned short R, int Init, int * VL, int * VR);
extern "C" __declspec(dllexport) bool Algo4(unsigned short S, unsigned short L, unsigned short R, int Init, int * VL, int * VR);
extern "C" __declspec(dllexport) bool Algo5(unsigned short S, unsigned short L, unsigned short R, int Init, int * VL, int * VR);
extern "C" __declspec(dllexport) bool Algo6(unsigned short S, unsigned short L, unsigned short R, int Init, int * VL, int * VR);
extern "C" __declspec(dllexport) bool Algo7(unsigned short S, unsigned short L, unsigned short R, int Init, int * VL, int * VR);

#endif // MYALGO_H
```

```
cmake_minimum_required(VERSION 3.5)
project(MyAlgo)

set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -std=c++11")

set(CMAKE_SHARED_LIBRARY_PREFIX "")

set(SOURCE_FILES MyAlgo.cpp MyAlgo.h)
add_library(MyAlgo SHARED ${SOURCE_FILES})
```

Algoritmanızdan Beklenenler:

- Pist ve yol renkleri deęişkenlik gösterebilir. Yol siyah, pist beyaz, yol beyaz, pist siyah olabilir. Algoritmanız buna göre sonuç verebilmelidir.
- Algoritmanız Acemi-1, Acemi-2, Acemi-3, ve Özel-1 pistlerinin tamamında başarıyla sonuca ulaşmalıdır.
- Usta-1, Usta-2, ve Usta-3 pistlerini başarıyla tamamlayan algoritmalar ekstra puan alacaktır.
- Algoritmanızı verilen 7 araç tipi için de denemelisiniz. Algoritmanız ilk araç olan Tosbaęa ile yukarıda verilen pistleri tamamlamalıdır. Algoritmanızın tüm araçları başarıyla sonuca ulaştırması ekstra puan getirecektir.
- Kodlarınızı **github** hesabınız altında oluşturacağınız robot projesi altında versiyonlayınız.
- Öğrenci numaranızın son hanesine göre algoritmanızı Algo1 ~ Algo7 'den hangisine yazacağınızı belirleyiniz. Son hane 1 ~ 7 sayılarından biriye doğrudan karşılık gelen algoritma fonksiyonunu yazınız. Son hane 7'den büyük ya da 0 ise, bundan önceki hanelerde 1~7 aralığındaki en yakın sayı seçilir.

Öğr. No	Son Hane	Algoritma
1		Algo1
2		Algo2
3		Algo3
4		Algo4
5		Algo5
6		Algo6
7		Algo7

- DLL dosyanızı çağırarak robot kontrolü yaptığınızda, kontrol algoritmanızın kaç döngü yaparak pisti tamamladığı simülatör tarafından sayılacaktır. En az sayıda döngü yaparak pisti tamamlamak amaçlanmaktadır. Bu yüzden algoritmanızı iyileştirmeye çalışırken döngü sayınızı da minimize etmeye çalışınız. Döngü sayısı ters orantılı olarak ödev notunuza etki edecektir.
- Özel 1 isimli pist önümüzdeki haftalarda ders materyalleri sayfasından sizlerle paylaşılacaktır.