



## طراحی و ساخت یک ربات بسترپویا برای پیاده‌سازی تکنیک عدم برخورد با موانع بر اساس سیستم فازی و بینایی ماشین

علی راوری<sup>1</sup>، مهدی طالع ماسوله<sup>2\*</sup>

1- دانشجوی کارشناسی ارشد، مهندسی مکترونیک، پردیس علوم و تحقیقات سمنان، دانشگاه آزاد اسلامی، سمنان، ایران  
2- استادیار، آزمایشگاه تعامل انسان و ربات، دانشکده علوم و فنون نوین، دانشگاه تهران، تهران، ایران  
\* m.t.masouleh@ut.ac.ir, 14395-1561, تهران

### اطلاعات مقاله

مقاله پژوهشی کامل  
دریافت: 16 دی 1394  
پذیرش: 20 بهمن 1394  
ارائه در سایت: 01 اردیبهشت 1395  
کلید واژگان:  
ربات بستر پویا  
کینکت  
این‌سی‌وی  
آردوینو  
الگوریتم فازی

### چکیده

این مقاله طراحی مکانیک یک ربات بسترپویا با نام TL-PR را ارائه می‌دهد. دو الگوریتم مجزا برای سیستم عدم برخورد با موانع بر روی ربات پیاده‌سازی شده است. برد کنترلی به کار گرفته شده در ربات آردوینو است که یک برد منبع باز می‌باشد. برای دریافت تصاویر مورد استفاده در بخش عدم برخورد و شناسایی اجسام از حسگر کینکت در ربات استفاده شده است. برای طراحی ساختار ربات از یک سیستم ساده و ابتکاری ارزان قیمت بهره برده شده است. برای سیستم عدم برخورد دو روش پیاده‌سازی شده است. یک روش بر اساس حسگر مافوق صوت است. در اطراف ربات تعداد 5 عدد سنسور مافوق صوت کار گذاشته شده است. برای استفاده از سنسورها از متد فازی استفاده شده و قوانین مربوط به آن در متلب پیاده‌سازی شده است. روش دوم، بر مبنای الگوریتم پردازش تصویر است. یک کینکت در بالای ربات برای سیستم عدم برخورد با موانع نصب شده است. این متد دوم در محیط نرم‌افزار ویژوال استودیو و با استفاده از کتابخانه این‌سی‌وی اجرا شده است. این ربات پلتفرم مناسبی برای ربات‌های خانگی و خدماتی است. به دلیل نصب لپ تاپ بر روی ربات، امکان پیاده‌سازی انواع الگوریتم‌های پیشرفته کنترلی و پردازش تصویر وجود دارد.

## Design and development of a mobile robot for implementing obstacle avoidance techniques based on fuzzy logic and vision

Ali Ravari<sup>1</sup>, Mehdi Tale Masouleh<sup>2\*</sup>

1-Department of Electrical and Computer Engineering, Semnan Science and Research Branch, Islamic Azad University Semnan, Iran  
2- Human and Robot Interaction Laboratory, Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran  
\* P.O.B. 14395-1561, Tehran, Iran, m.t.masouleh@ut.ac.ir

### ARTICLE INFORMATION

Original Research Paper  
Received 06 January 2016  
Accepted 09 February 2016  
Available Online 20 April 2016

**Keywords:**  
Mobile robot  
Kinect  
OpenCV  
Arduino  
Fuzzy method

### ABSTRACT

This article presents the mechanical design process of a mobile robot which is named TL-PR. Two separate algorithms are applied for obstacle avoidance purpose that are experimentally implemented on the proposed robot. The control board Arduino which is used for the under study robot is an open source board. In order to receive the images that are used for obstacle detection and obstacle avoidance a Kinect sensor is installed in the proposed robot. The structure of TL-PR is a creative, simple and low cost structure. Two methods are implemented on the proposed robot for obstacle avoidance. The first one is based on ultrasonic sensor. Five ultrasonic sensors are set around the proposed robot structure. The fuzzy control is used to manage the output data of the ultrasonic sensors and the rules of the fuzzy control are set on the matlab software. The second method which is used for obstacle detection and avoidance is based on image processing algorithm. A Kinect sensor is set on the top of the robot structure for image processing to detect the obstacles. The second method consists of processing the visual studio software and it run based on the OpenCV library. The proposed robot is a desirable platform for the @home robots. The laptop, which is set on the robot made the robot compatible for implementing the various control and image processing algorithms.

### 1- مقدمه

های خانگی<sup>2</sup> برای کمک به افراد کم‌توان و معلول و انجام کارهای منزل، رویکرد بسیار فعالی دارند. هدف از ساخت این ربات‌ها کمک به افراد سالمند، بیماران و معلولین بوده است. این نوع از ربات‌ها به دلیل وجود حسگرها، دوربین‌ها و تجهیزات مناسب به سادگی بدون برخورد با موانع موجود در منزل، بین وسایل حرکت کرده و به نقطه مورد نظر خود می‌رسند. این گونه از

ربات‌ها در زندگی امروزه انسان نقش گسترده‌ای را ایفا می‌کنند. در این میان ربات‌های پایه پویا<sup>1</sup> به دلیل تنوع در ساخت و پیاده‌سازی انواع الگوریتم‌ها در میان پژوهشگران محبوبیت زیادی دارند. ساخت رباتی که در زندگی انسان نقش مؤثری داشته باشد، یکی از اهداف محققان است. به عنوان نمونه ربات-

2- At home robot

1- Mobile robot

Please cite this article using:

A. Ravari, M. Tale Masouleh, Design and development of a mobile robot for implementing obstacle avoidance techniques based on fuzzy logic and vision, *Modares Mechanical Engineering*, Vol. 16, No. 4, pp. 271-280, 2016 (in Persian)

برای ارجاع به این مقاله از عبارت ذیل استفاده نمایید:

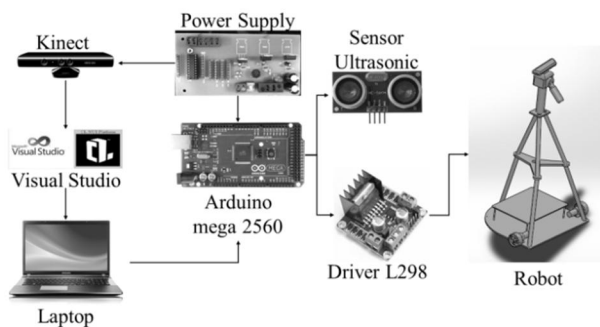


Fig. 1 Block diagram of TL-PR robot

شکل 1 بلوک دیاگرام کلی قطعات ربات

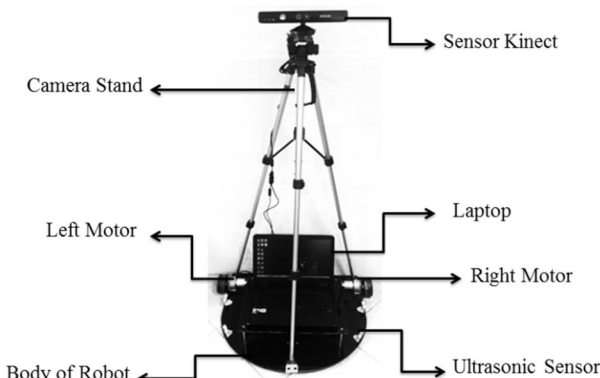


Fig. 2 TL-PR robot, designed in Human and Robot Interaction Laboratory

شکل 2 ربات تی‌ال‌پی‌آر ساخته شده در آزمایشگاه تعامل انسان و ربات دانشگاه تهران

باشد. به دلیل زمان مورد نیاز محاسبه این مرحله، این روش در محیط‌هایی با جزئیات و موانع زیاد خیلی مناسب نمی‌باشد.

از این روش مذکور تنها برای تعیین مسیر به صورت غیر لحظه‌ای مناسب است و برای پرهیز از مانع به صورت بلادرنگ<sup>12</sup> که یکی از موضوعات مهم در محیط‌های پویا می‌باشد، کافی به نظر نمی‌رسد. اگر یک مانع متحرک ظاهر شود مسیر می‌بایست تغییر کند. این موضوع باعث می‌شود که مسیر حرکت دوباره محاسبه شود که ممکن است نسبت به حرکت مانع زمان زیادی بگیرد. بنابر این واکنش ربات خیلی مناسب نخواهد بود. نقطه ضعف دیگر الگوریتم‌های سراسری این است که نیاز به دانش دقیقی از محیط دارند. اینکه یک مانع به فضایی که قبلاً خالی بوده است وارد شده باشد، مانند یک مانع متحرک باعث محاسبه دوباره اطلاعات ورودی می‌شود. در نهایت یک ربات با الگوریتم ناوبری سراسری تنها به اهدافی می‌رسد که از قبل در نقشه شناخته شده باشند [6].

روش محلی نیازمند هیچ مرحله پیش محاسبه سنگینی قبل از آغاز جستجو نیست. روش‌های محلی اطلاعات کنونی حسگر را به صورت بلادرنگ بدست می‌آورند تا ربات را از موانع دور نگه دارند. به دلیل آنکه روش‌های محلی هیچ‌گاه قسمت‌های مختلف تصویر را از پیش محاسبه نمی‌کنند، استفاده از آن‌ها بهتر به نظر می‌رسد. هر دو روش سراسری و محلی دارای مزایا و معایبی هستند که برای کاربردهای مختلف مناسب می‌باشند [7].

در این مقاله از روش محلی برای دریافت تصاویر استفاده شده است. حسگر کینکت در بالای ربات نصب شده و اطلاعات را به صورت بلادرنگ به سیستم پردازش ربات ارسال می‌کند.

به جز روش‌های مبتنی بر پردازش تصویر، الگوریتم‌های دیگری نیز برای

ربات‌ها از سطح هوشمندی بالایی برخوردار می‌باشند. بیشتر این ربات‌ها برای این که عملکرد قابل قبولی داشته باشند الگوریتم‌های مسیریابی<sup>1</sup>، شناسایی اشیاء<sup>2</sup> عدم برخورد با موانع<sup>3</sup> را اجرا می‌کنند. در شکل 1 تمامی بخش‌های سخت‌افزاری ربات از قبیل برد کنترلی، سنسورها، درایور، برد تغذیه، کینکت به همراه نحوه اتصال آن‌ها مشخص شده است [1].

انواع دیگری از این گروه ربات، ربات‌های خدمت‌رسان<sup>4</sup> هستند. این نوع از ربات هم همانند ربات خانگی به منظور خدمت به انسان در موقعیت‌های خاصی ساخته شده است. این ربات باید قابلیت شناسایی افراد و تعقیب آن‌ها را در محیط‌هایی با موانع زیاد داشته باشد. هر دوی این سری از ربات‌ها دارای مزایا و معایبی می‌باشند [2].

در این مقاله بر روی رباتی تمرکز شده که پیکربندی مناسبی برای پیاده‌سازی انواع الگوریتم‌های پردازش تصویر، ترکیب اطلاعات حسگر<sup>5</sup>، کنترل و غیره را داراست. این ربات با استفاده از حسگر کینکت، حسگرهای مافوق صوت و سخت‌افزار منبع باز<sup>6</sup> زمینه مناسبی را برای پیشرفت الگوریتم‌های هوشمند ایجاد می‌کند.

در ربات ارائه شده در این مقاله، برای دریافت تصاویر از محیط اطراف از کینکت استفاده می‌شود. این اطلاعات تصویر که به صورت خام از تصویر رنگی<sup>7</sup> گرفته می‌شوند، با کمک درایور مربوطه<sup>8</sup> به اطلاعات مناسبی برای کینکت به عنوان تصویر ورودی تبدیل می‌شوند [3].

بعد از این که تصویر رنگی توسط توابع کتابخانه اپن‌سی‌وی<sup>9</sup> پردازش شد، دستورات کنترلی به ربات ارسال می‌شود. کتابخانه اپن‌سی‌وی یکی از پر کاربردترین توابع در بحث پردازش تصویر و ردیابی اشیاء است. در این مقاله برای نوشتن الگوریتم‌های مورد استفاده از محیط برنامه‌نویسی ویژوال استودیو استفاده شده است. ارتباط سریال بین کامپیوتر و برد کنترلی از طریق پورت سریال و دستورات مربوطه انجام می‌شود [4].

یکی از مباحث مهم پیرامون ربات‌های متحرک سیستم عدم برخورد ربات با موانع است. ربات باید بتواند در محیط‌هایی با موانع مختلف به راحتی بدون برخورد با موانع مسیر خود را پیدا کند. از این رو در این مقاله علاوه بر معرفی بخش مکانیک ربات، سعی بر آن شده تا دو روش عدم برخورد با مانع نیز بیان گردد. در شکل 2 ربات مورد نظر به همراه اجزای تشکیل دهنده مشخص شده است.

برای پیاده‌سازی الگوریتم‌های عدم برخورد با موانع روش‌های مختلفی وجود دارد. روش‌هایی که بر مبنای بینایی هستند، به طور معمول به دو دسته کلی تقسیم می‌شوند. یک روش محلی<sup>10</sup> و دیگری روش سراسری<sup>11</sup> است [5]. در روش سراسری دوربین بر روی خود ربات نصب نیست و در محل دیگری که بر محیط دید مناسبی دارد، نصب می‌شود. به طور کلی یک الگوریتم سراسری نتایج مطلوب‌تری از نوع محلی بدست می‌دهد. دلیل این مسأله این است که این نوع الگوریتم‌ها توانایی دریافت تمام اطلاعات جزئی درباره محیط حرکت ربات را دارند. نقطه ضعف روش سراسری این است که نیازمند یک مرحله پیش محاسبه سنگین قبل از آغاز جستجوی مسیر می‌-

- 1- Path planning
- 2- Object detection
- 3- Obstacle avoidance
- 4- Present robots
- 5- Sensor fusion
- 6- Open source
- 7- RGB
- 8- CL nui
- 9- Open CV
- 10- Local
- 11- Global

12- Real time

کرد. در این ربات‌ها معمولا از دوربین‌های مختلفی به عنوان ورودی تصویر استفاده می‌شود. این ربات‌ها از حسگرهای متفاوتی برای کاربردهای مورد نظر خود استفاده می‌کنند. از نظر مکانیکی این ربات‌ها ارتفاع نسبتا بلندی دارند. دلیل آن هم نیاز به دوربین در یک ارتفاع مناسب برای پیاده‌سازی مناسب الگوریتم‌های پردازش تصویر است. به همین منظور این ربات‌ها در چند طبقه برای قرار گیری تجهیزات الکترونیک ساخته می‌شوند.

در ربات‌هایی که ارتفاع بلندی دارند، ممکن است در حین حرکت لرزش-هایی در قسمت‌های بالایی ربات به وجود بیاید. به طور معمول یک دوربین را در بخش‌های بالایی ربات قرار می‌دهند. این لرزش‌ها در روند پردازش تصویر ممکن است اختلال ایجاد نمایند. برای رفع این مشکل از یک روش ابتکاری در این مقاله استفاده شده است.

مدل ساختاری که برای این ربات در نظر گرفته شده است، یک ساختار افتراقی<sup>4</sup> است. این سیستم برای ربات‌هایی که برای کارهای تحقیقاتی طراحی و ساخته می‌شوند، بسیار مناسب است. به دلیل سادگی در طراحی و هزینه پایین در ساخت نسبت به دیگر ساختارها، این مدل از ربات‌ها برای انجام پروژه‌های تحقیقاتی بسیار مناسب هستند. مکانیزم افتراقی دارای دو موتور متحرک و یک هرزگرد به منظور ایجاد تعادل است [11].

برای جلوگیری از ایجاد لرزش، در این ربات به جای ایجاد چند طبقه، از یک سه پایه عکاسی از جنس فیبر کربن استفاده شده است. این پایه دوربین به دلیل وجود شکل ظاهری شبیه به خرپا تا حد بسیار زیادی لرزش بوجود آمده را از بین می‌برد. دوربین کینکت در بالای این سه پایه قرار خواهد گرفت. استفاده از سه پایه عکاسی این امکان را می‌دهد تا به راحتی و بدون نیاز به تغییر در جزییات بدنه ربات ارتفاع دوربین را تغییر داد. تغییر ارتفاع باعث می‌شود تا الگوریتم‌های مختلفی را بتوان بر روی ربات پیاده‌سازی نمود. بدنه این ربات از جنس پلکسی گلاس است. از ویژگی‌های بارز پلکسی که استفاده از آن را مقرون به صرفه کرده است، می‌توان به مواردی همچون وزن کم، مقاومت بالا، برش راحت و قیمت مناسب اشاره نمود. این ربات در دو طبقه ساخته شده است. طبقه اول ربات برای نصب موتورها و درایور موتور، سنسورهای مافوق صوت، برد کنترلی، برد تغذیه طراحی شده است. در طبقه دوم لپ‌تاپ قرار خواهد گرفت. الگوریتم‌های پردازش تصویر به دلیل حجم بالای پردازش، نیاز به سخت‌افزار مناسبی برای اجرا دارند. به همین دلیل برای اطمینان از اجرای سریع الگوریتم‌ها از یک لپ‌تاپ استفاده شده است. این لپ‌تاپ برای ارتباط با کینکت و ارسال دستورات کنترلی به برد آردوینو<sup>5</sup> است.

به دلیل نوع خاص بدنه، موتورهایی که به عنوان نیروی محرک ربات استفاده می‌شوند، باید از سرعت کمی برخوردار باشند ولی در عوض قدرت و گشتاور مناسبی را ارائه دهند. به همین منظور از موتوری با سرعت 25 دور در دقیقه<sup>6</sup> با گیربکس فلزی استفاده شده است. در سیستم افتراقی برای نیروی محرک ربات از دو موتور استفاده می‌شود. برای ایجاد تعادل در ربات یک هرزگرد در جلوی ربات قرار داده شده است. هرزگردهای این ربات از نوع بلبرینگ چشمی هستند. در کل این ربات در سه نقطه با زمین در تماس است. در این ربات از چرخ‌های پلاستیکی با قطر 10 سانتی‌متر و پهنای 4 سانتی‌متر استفاده شده است. در شکل 3 بلوک دیاگرام مربوط به ارتباط اجزای سخت‌افزار ربات و سیستم عدم برخورد روش پردازش تصویر و روش

عدم برخورد با موانع وجود دارند. در این روش‌ها به طور معمول از حسگرهای مختلفی برای شناسایی موانع استفاده می‌شود. از این قبیل حسگرها می‌توان به سنسور مافوق صوت<sup>1</sup>، مادون قرمز<sup>2</sup>، لیزر اسکنر اشاره کرد. از روش‌های دیگر عدم برخورد می‌توان به روش میدان پتانسیل اشاره کرد [8].

برای انجام الگوریتم‌های پردازش تصویر روش‌ها و کتابخانه‌های متفاوتی وجود دارد. یکی از بهترین کتابخانه‌ها در این زمینه اپن‌سی‌وی می‌باشد. این کتابخانه در پیاده‌سازی الگوریتم‌های بسیاری در زمینه بینایی ماشین استفاده می‌شود. از آنجا که ویدئوها مجموعه‌ای از تصاویر هستند، روش‌های بینایی ماشین قابلیت اضافه شدن و پردازش ویدئوها را هم دارند. برای انجام پروژه-هایی همچون تشخیص چهره، شناسایی اشیاء و تشخیص حرکت در تصویر، اپن‌سی‌وی الگوریتم‌های مورد نیاز برای استفاده را فراهم می‌کند [9].

در این مقاله، به کنترلی برپایه مفاهیم فازی پرداخته می‌شود. کنترل فازی برای حل مسائل دنیای واقعی توسعه یافته است. منطق فازی، تنظیم فازی، مدل سازی فازی و روش‌های کنترل فازی تماما شکل گرفته توسط انسان بوده و به طور موضوعی توسط احساس انسانی معرفی می‌شوند. اگر تعبیر فازی صحیح باشد و ریاضیات تئوری فازی عمل کنند، آن وقت است که فرد می‌تواند قابلیت حل مشکلات دنیای واقعی را پس از کامل شدن عملکرد فازی در محیط فازی داشته باشد و سپس آن را به دنیای اصلی خود یعنی فضای غیر فازی برگرداند. این همان چیزی است که تحت عبارات فازی کردن، فازی عمل کردن و غیر فازی کردن مطرح می‌شود.

بیشتر سیستم‌های کنترل فازی سیستم‌هایی هستند که مدل فازی آنها یا کنترلر آنها بر اساس یکسری قوانین اگر-آنگاه<sup>3</sup> بر پایه تجربه و دانش فرد طراح بیان می‌شود. اولین نوع از سیستم کنترل فازی آن است که منطق فازی مستقیما تمام فعالیت‌های کنترلی را در دست بگیرد و کاملا جایگزین سیستم کنترل کلاسیک شود. نوع دیگر از آن همراه سیستم کنترل کلاسیک و در کنار آن قرار می‌گیرد و به منظور بهبود عملکرد کنترل بخشی از سیستم کنترل ترکیبی را در بر می‌گیرد [10].

این ربات از مکانیزم ابتکاری و کم هزینه‌ای بهره می‌برد. ربات به گونه‌ای طراحی شده است که دانشجویان علوم رباتیک به راحتی بتوانند برای اجرای پروژه‌های خود از آن استفاده کنند. به عنوان بخش الکترونیک ربات از سخت‌افزار متن باز استفاده شده است. بر روی این ربات دو الگوریتم مختلف برای سیستم عدم برخورد ربات با موانع پیاده‌سازی شده است. الگوریتم اول بر مبنای بینایی ماشین و پردازش تصویر است. الگوریتم دوم بر مبنای سیستم فازی است.

در این مقاله در بخش اول مکانیک ربات بررسی شده است. ساختار ربات از نظر ظاهری و دلیل استفاده از چنین سیستمی در این بخش توضیح داده شده است. در بخش بعدی الکترونیک ربات مد نظر است. استفاده از سخت‌افزار منبع باز و حسگرهای مورد استفاده برای اجرای دستورات از نکات مهم این بخش است. در بخش سوم هم الگوریتم مورد استفاده برای سیستم عدم برخورد بر مبنای بینایی ماشین و روش فازی برای استفاده از حسگرهای مافوق صوت توضیح داده شده است.

## 2- مکانیزم ربات

برای ساخت ربات‌هایی از این قبیل، ساختارهای مختلفی در دنیا موجود است. از این قبیل ربات‌ها می‌توان به ربات‌های خانگی یا ربات‌های خدماتی اشاره

4- Differential drive  
5- Arduino  
6- RPM

1- Ultrasonic  
2- Ir  
3- If-then

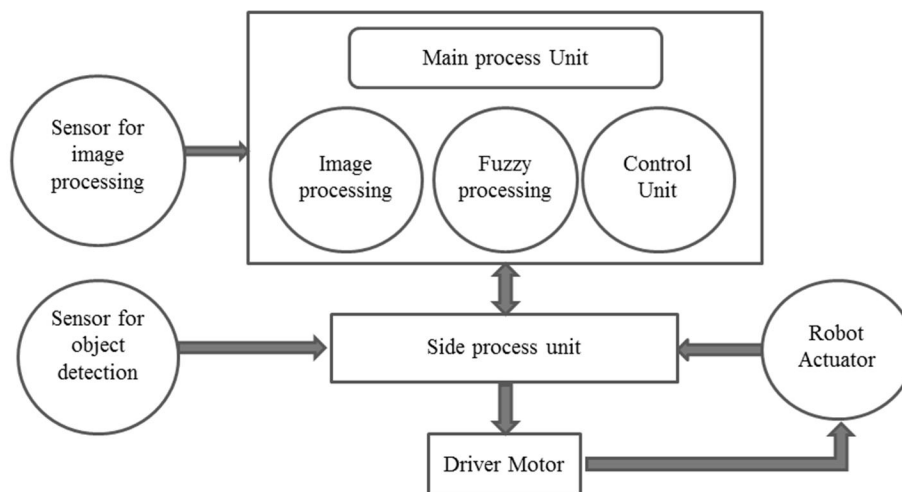


Fig. 3 Block diagram for image processing system with fuzzy control

شکل 3 بلوک دیاگرام سیستم پردازش تصویر به همراه کنترل فازی

فازی مشخص شده است.

صنعتی استفاده شود [13].

### 3- الکترونیک ربات

در این بخش تجهیزات مورد استفاده در الکترونیک ربات همچون برد کنترلی، حسگرها و درایور موتور مورد بررسی قرار می‌گیرد. هر کدام از این موارد به صورت جداگانه بررسی شده است و سعی بر آن بوده است تا معایب و مزایای هر کدام بررسی شود. شکل 4 تصویر خروجی برنامه را نشان داده است.

#### 1-3- برد کنترلی آردوینو

در این بخش به جای طراحی یک برد کنترلی صرفاً مخصوص ربات مذکور، از یک برد منبع باز استفاده می‌شود. مزیت استفاده از برد آردوینو این است که اگر تغییراتی در مشخصات فنی ربات ایجاد گردد، دیگر نیاز به طراحی مجدد برد کنترلی و پرداخت هزینه اضافی نیست. در بین تمامی بردهای منبع باز موجود در بازار، سخت‌افزار مبتنی بر آردوینو یکی از بهترین این بردها می‌باشد. به عنوان برد کنترلی ربات از برد آردوینو مگا 2560 استفاده شده است. از مزایای این برد می‌توان به 54 پین ورودی- خروجی دیجیتال، 16 پین ورودی آنالوگ، کریستال اسیلاتور 16 مگا هرتز، 4 واسط سریال<sup>1</sup> و اتصال ساده به وسیله پورت سریال یو اس بی<sup>2</sup> نام برد [12].

#### 2-3- ماژول درایور

همانطور که گفته شد از دو موتور برای این ربات استفاده شده است. بنابر این به درایوری احتیاج است که توانایی درایو کردن این موتورها را داشته باشد. در این پروژه از ماژول درایور<sup>3</sup> بهره برده شده است. این درایور به دلیل وجود پل اچ<sup>4</sup> در داخل خود توانایی درایو دو موتور را دارد. اگر صرفاً از خود آی‌سی<sup>5</sup> درایور استفاده شود، نیاز به یک مدار مناسب برای اتصال به موتور، تغذیه آی سی و محافظت از جریان‌های بازگشتی می‌باشد. اما با استفاده از این ماژول دیگر نیاز به طراحی مدار نیست و یک مجموعه کامل، ارزان و در دسترس در اختیار کاربر قرار دارد. این درایور قابلیت درایو موتورهایی با جریان حدود 3 آمپر را داراست. برای موتوری با جریان کشی بالاتر بهتر است از درایورهای

#### 3-3- موتور

وزن ربات مذکور حدود 10 کیلوگرم است، بنابراین نیاز به موتورهایی دارد که بتواند این وزن را تحمل نماید. به طور معمول در چنین ربات‌هایی به دلیل پیاده‌سازی الگوریتم‌های پیچیده، سرعت خیلی بالا در حرکت ربات مورد نیاز نمی‌باشد. این ربات‌ها باید حرکت نرم و آهسته‌ای داشته باشند تا در اجرای الگوریتم‌ها مشکلی پیش نیاید. به همین دلیل در این ربات از موتورهای 12 ولت با سرعت 25 دور در دقیقه استفاده شده است. برای راه اندازی این موتور هم از یک باتری 12 ولتی با 2.2 آمپر استفاده شده است.

این نوع موتور از نوع موتور دی سی بوده و برای ایجاد یک قدرت مناسب در حرکت از گیربکس‌های فلزی در این موتور استفاده شده است. گیربکس‌های این موتور همگی از نوع دندانه راست هستند. به دلیل وجود چنین گیربکس‌هایی، موتور از گشتاور مناسبی برای حرکت ربات برخوردار است. برای دریافت بازخورد از موتور انکودر جداگانه استفاده شده است.

#### 4-3- برد تغذیه

به دلیل تفاوت ولتاژهای کاری تجهیزات استفاده شده در ربات مذکور یک برد تغذیه چند منظوره برای تامین ولتاژهای مورد استفاده در ربات طراحی شده است. برد طراحی شده دارای ولتاژهای 5 ولت، 9 ولت و 3.3 ولت است. سه مدل رگولاتور<sup>6</sup> مختلف برای تولید این ولتاژها به همراه خازن‌های محافظتی به کار رفته است. برای ایجاد ولتاژهای مد نظر از رگولاتورهای ال اف 33، 7805 و 7809 استفاده شده است. این برد به گونه‌ای طراحی شده است که قابلیت استفاده در دیگر ربات‌ها را نیز دارد.

#### 5-3- سنسور مافوق صوت

برای عدم برخورد ربات با موانع روش‌های متفاوتی وجود دارد. یکی از این روش‌ها استفاده از حسگرهای مافوق صوت می‌باشد. در این مقاله از حسگرهای مافوق صوت مدل sr04 استفاده شده است.

این حسگرها با برد آردوینو به راحتی قابل استفاده می‌باشند. فاصله مناسب برای عدم برخورد ربات با موانع، با استفاده از این حسگر قابل تنظیم است. در این ربات 5 عدد از حسگرهای مافوق صوت استفاده شده است. این

1- Uart  
2- usb  
3- L298  
4- H-bridge  
5- IC

6- Regulator

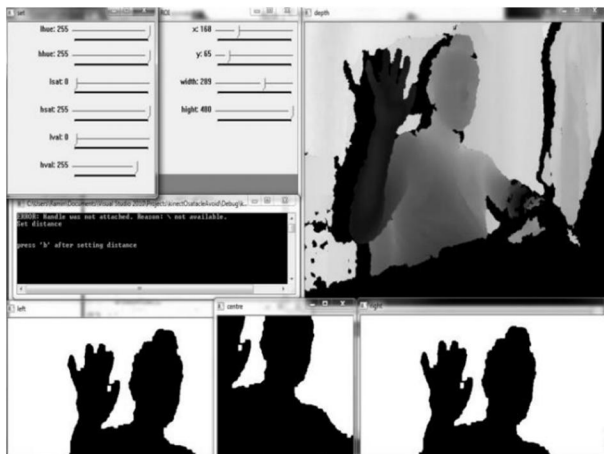


Fig. 4 Run algorithm1 and obtained depth image from Kinect

شکل 4 اجرای الگوریتم 1 و دریافت تصویر عمق به دست آمده از کینکت

همین دلیل اگر مانعی به ربات خیلی نزدیک باشد و فاصله آن با ربات کمتر از 45 سانتی متر باشد، قابل تشخیص نیست. در تصویر عمق<sup>8</sup> کینکت هرچه جسم نزدیک‌تر باشد، تیره‌تر و هرچه دورتر باشد روشن‌تر می‌باشد.

مانع می‌تواند به عنوان رنگ سیاه در تصویر شناسایی شود. این موضوع با استفاده از توابع موجود در این سی‌وی انجام می‌شود. این توابع برای تشخیص رنگ در تصویر استفاده می‌شود، بنابراین شناسایی تصویر سیاه و سفید در تشخیص مانع امکان‌پذیر است [16].

باید به این نکته توجه کرد که ربات دقیقاً تصویر جلوی خود را می‌بیند و اطلاعاتی از مناطق اطراف خود ندارد. این مشکل با استفاده از توابع موجود در این سی‌وی قابل حل است. با تنظیم منطقه مورد نظر<sup>9</sup>، تنها به پردازش بخشی از کل تصویر نیاز است. چون تنها بخشی از تصویر نیاز به پردازش دارد، زمان کمتری برای اجرا لازم است. طبق الگوریتم استفاده شده در این مقاله، تصویر به سه بخش میانی، چپ و راست تقسیم می‌شود. وقتی برنامه اجرا می‌شود، چند پنجره در صفحه لپ‌تاپ ظاهر می‌گردد. یکی از این پنجره‌ها مربوط به تنظیم مقادیر  $X$  و  $Y$  می‌باشد. این دو مقدار مربوط به اندازه صفحه مرکز تصویر است. در واقع با تنظیم مقادیر  $X$  و  $Y$  آن بازه‌ای از تصویر که الگوریتم شناسایی مانع بر روی آن انجام می‌شود، مشخص خواهد شد. تعیین مقدار صحیح این مقادیر در عملکرد صحیح این الگوریتم حائز اهمیت می‌باشد.

اگر این دو مقدار به درستی انتخاب نشوند ممکن است در بعضی نقاط عدد به دست آمده از تصویر منفی شده و این مسأله برای ربات غیر قابل قبول خواهد بود. دیگر پنجره‌های ایجاد شده در صفحه نمایش بیانگر تصاویر میانی، راست، چپ و عمق دوربین کینکت می‌باشند. بعد از اجرای برنامه و حرکت ربات سه پنجره‌ای که برای کاربر بسیار مهم است، مرکز، راست و چپ می‌باشند. موانع به صورت رنگ مشکی در این تصاویر دیده می‌شوند. الگوریتم 1 شبه کد بخش پردازش تصویر را نشان می‌دهد. در الگوریتم 2 نحوه دریافت هر کاراکتر و عکس العمل ربات دیده می‌شود. تصمیم‌گیری برای گردش به چپ یا راست بسیار مهم است. اگر مانعی بر سر راه ربات نباشد، ربات رو به جلو حرکت می‌کند. اگر مانعی بر سر راه ربات قرار بگیرد، نحوه چرخش به سمت راست یا چپ و عبور از مانع مهم است. اگر مانع در تصویر سمت چپ دیده شود، به صورت یک جسم مشکی رنگ در پنجره چپ دیده می‌شود و در این صورت اگر مانع دیگری نباشد ربات به سمت راست می‌چرخد. به همین

حسگرها دارای 4 پایه هستند. دو پایه 5 ولت و زمین<sup>1</sup> برای تغذیه حسگر و دو پایه تریگ<sup>2</sup> و اکو<sup>3</sup> هم برای دریافت و تنظیم پهنای پالس<sup>4</sup> مورد نیاز می‌باشند. طبق نتایج بدست آمده، به طور معمول این مدل حسگر تا حدود 80 سانتی متر اهداف را با دقت مناسبی شناسایی می‌کند [14].

### 3-6- حسگر کینکت

در تمام روش‌های معرفی شده برای پردازش تصویر همیشه نیاز به یک دوربین یا حسگر برای دریافت تصویر ورودی می‌باشد. برای دریافت تصویر دوربین‌های مختلف با کارایی متفاوت وجود دارند. اگر نیاز به پردازش عمق تصویر در الگوریتم مورد نظر دیده می‌شود باید از دوربین استریو استفاده شود. یکی از دوربین‌هایی که امروزه پرکاربرد می‌باشد حسگر کینکت است. حسگر کینکت با قابلیت‌هایی که دارد نیاز به استفاده از دوربین استریو را برطرف می‌کند.

حسگر کینکت برای موقعیت‌یابی در بالا یا جلوی ربات نصب می‌شود. به دلیل وجود یک موتور در داخل بدنه کینکت، این وسیله قابلیت حرکت بالا و پایین را به صورت محدود دارد. این وسیله بر پایه یک دوربین با دو عدس لنز و یک درگاه مادون قرمز است که درگاه مادون قرمز و یکی از دوربین‌ها برای تشخیص عمق تصاویر است و دوربین دیگر هم که یک دوربین رنگی است، برای تشخیص تصاویر می‌باشد. از دیگر تجهیزات موجود در حسگر مذکور می‌توان به وجود یک شتاب سنج سه محوره و آرایه‌ای از میکروفن‌ها و یک موتور اشاره نمود. در جلوی بدنه کینکت میکروفن، فرستنده مادون قرمز، ال-ای‌دی چند رنگ، دوربین وی‌جی‌ای<sup>5</sup> و دوربین مادون قرمز قرار دارد. اولین قدم برای انجام این پروژه دسترسی به اطلاعات کینکت و ارسال اطلاعات دریافتی آن به ویژوال استودیو می‌باشد. این کار با استفاده از درایور سی‌ال<sup>6</sup> و بسته توسعه نرم‌افزار<sup>7</sup> کینکت انجام می‌شود [15]. یکی از تفاوت‌های عمده حسگر کینکت با دیگر دوربین‌ها در این است که برای دریافت تصویر حتماً می‌بایست مقدمات این کار را فراهم نمود. بدین منظور لازم است ابتدا درایورهای لازم بر روی سیستم نصب شود. درایور سی‌ال به داده‌های حسگر عمق کینکت دسترسی دارد و تصویر رنگی در کتابخانه این‌سی‌وی را فراهم می‌کند. این درایور با استفاده از دستور مربوطه در توابع کتابخانه این‌سی‌وی، اطلاعات را از سنسور کینکت دریافت می‌کند.

ارزش و مقدار پیکسل خاص در یک موقعیت در تصویر رنگی به فاصله جسم از دوربین بستگی دارد. داده‌های سنسور عمق کینکت با استفاده از توابع موجود در این‌سی‌وی به تصویر رنگی تبدیل می‌شوند. این توابع اطلاعات سنسور عمق را به 3 کانال 8 بیتی تصویر رنگی تبدیل می‌کنند. توابع موجود در این‌سی‌وی اطلاعات خام تصویر را از کینکت دریافت می‌کند. در مرحله بعدی این تصاویر به لپ‌تاپ برای پردازش منتقل خواهد شد.

### 4- شناسایی موانع با استفاده از بینایی ماشین

شناسایی موانع با روش‌های زیادی انجام می‌شود. یکی از این روش‌ها، تشخیص مانع بر اساس پردازش تصویر و بینایی است. کینکت تشخیص اجسام را از فاصله‌ی 45 سانتی متری خود شروع می‌کند. بنابراین در فاصله بین 0 تا 45 سانتی متر امکان مشاهده و تشخیص چیزی وجود ندارد. به

1- GND  
2- Trig  
3- Eco  
4- PWM  
5- VGA(video graphic array)  
6- CL nui  
7- SDK (Software development kit)

8- Depth image  
9- Region of interest (ROI)

**Algorithm 2** The pseudo-code for Arduino code.

```

Initialize: char data
2: Initialize: void setup()
   Initialize: Serial.begin(9600)
4: pins 8,9,10,11 are Output
   Initialize: void loop()
6: if Serial.available() bigger than 0 then
   data ← Serial.read()
8:   if data == 'R' then
     pin 11 ← HIGH
10:    pin 10 ← LOW
     pin 9 ← LOW
12:    pin 8 ← HIGH
     if data == 'L' then
14:       pin 11 ← LOW
        pin 10 ← HIGH
16:       pin 9 ← HIGH
        pin 8 ← LOW
     if data == 'F' then
18:        pin 11 ← HIGH
         pin 10 ← LOW
20:        pin 9 ← HIGH
         pin 8 ← LOW
     if data == 'T' then
24:        pin 11 ← LOW
         pin 10 ← LOW
26:        pin 9 ← LOW
         pin 8 ← LOW
28:     end if
   end if
30: end if
end if
32: end if

```

**Algorithm. 2** The Pseudo-code for Arduino.

الگوریتم 2 شبه کد بخش کنترلی آردوینو.

اضافه شده است.

با استفاده از این کتابخانه، می‌توان به راحتی پورت مورد نظر را انتخاب کرد و از طریق آن دستورات مربوط به حرکت ربات را به آردوینو ارسال کرد. داده‌ها از طریق درگاه سریال فرستاده می‌شوند. این داده‌ها از نوع کاراکتر هستند. متغیرهای 'L' برای چپ، 'R' برای راست، 'F' برای مستقیم و 'T' برای توقف ربات ارسال می‌شوند. ارتباط سریال بین ربات، برد کنترلی و رایانه از طریق هدر فایل سریال پورت ایجاد شده است. همان طور که در الگوریتم 2 مشخص است، پایه‌های 8 الی 11 برد آردوینو برای کنترل ربات استفاده می‌گردد [12].

**6- الگوریتم فازی برای عدم برخورد**

ساختار کلی یک کنترل کننده با منطق فازی یا یک کنترلر فازی به طور خلاصه از سه قسمت پایه‌ای تشکیل می‌شود. در شکل 5 بلوک دیاگرام بخش فازی مشاهده می‌شود. ورودی‌های سیستم فازی در شکل 6 و خروجی‌ها در شکل 7 مشخص است. این سه بخش عبارتند از [18]:

- فازی سازی (بخشی که با ورودی‌ها ارتباط دارد).
- موتور استنتاج که مبتنی بر قواعد فازی است.
- غیر فازی سازی (بخشی که با خروجی‌ها ارتباط دارد)

**1-6- فازی سازی**

ماژول فازی سازی، مقادیر فیزیکی سیگنال‌های ورودی را به مقادیر فازی نرمالایز شده در بازه تغییرات ورودی به همراه یک سری توابع عضویت که

**Algorithm 1** The part of pseudo-code for image precessing algorithm.

```

1: if area bigger than 72 then
2:   cvSetImageROI(grey,cvRect(14, y, x - 14, h))
3:   [Left- > Image] ← grey
4:   check left area
5:   cvSetImageROI(grey,cvRect(w + x, y, 640 - (w + x +
   7), h))
6:   [Right- > Image] ← grey
7:   check right area if ((x-Left bigger than (x-14) bigger
   [Rightx]) and [Leftx] != 0)
8:   show on display RIGHT
9:   send on serial port command R
10:  else if x-Left- > x-14 < Right- > x or Left- >
   x == 0) then
11:   show on display LEFT
12:   send on serial port command L
13:   show on display FORWARD
14:   send on serial port F
15:  end if

```

**Algorithm. 1** Pseudo-code for image processing algorithm

الگوریتم 1 شبه کد برای روش پردازش تصویر

ترتیب اگر موانع در سمت راست تصویر باشد، ربات به سمت چپ خواهد چرخید. این عمل از طریق یک تابع مساحت<sup>1</sup> انجام می‌شود. تابع مذکور در الگوریتم 1 مشخص شده است. این تابع مساحت موانع را حساب می‌کند. حال اگر مساحت موانع از مساحت تصویری که در آن است بیشتر باشد بدین معنا است که موانع در آن سمت ربات قرار دارد. اما اگر موانع در هر دو تصویر راست و چپ حضور داشت، محاسبه نحوه چرخش ربات بسیار حائز اهمیت است. در این صورت، برنامه هر دو تصویر راست و چپ را از نظر مساحت موانع موجود در آن چک می‌کند. در هر سمت که مساحت موانع کوچک تر باشد، یعنی ربات در آن سمت با موانع کوچکتری رو به رو است. پس احتمال اینکه ربات بتواند از آن سمت عبور کند بیشتر می‌باشد [17].

تابع مساحت از یک مفهوم به نام لحظه<sup>2</sup> استفاده می‌کند. به منظور به دست آوردن یک معادله ریاضی برای مرکز شکل از مفهوم لحظه استفاده شده است. انواع مختلف لحظه در کتابخانه اپن‌سی‌وی با استفاده از پارامترهای معادله (1) محاسبه می‌شوند.

$$m(p, q) = \iint I(x, y) x^p y^q \quad (1)$$

در تابع  $m$  دو آرگومان  $p$  و  $q$  به عنوان توان برای  $x$  و  $y$  در نظر گرفته شده‌اند. تابع  $I$  مبین شدت<sup>3</sup> است، که در آن شدت برای پیکسل با مختصات  $(x, y)$  تعریف شده است.  $n$  تعداد پیکسل‌هایی است که شکل را تشکیل می‌دهند. تابع  $m$  برای محاسبه مرکز ثقل نقطه  $\bar{x}$  و  $\bar{y}$  برای یک شکل کافی است.

**5- ارتباط سریال**

داده‌های دریافتی از کینکت باید به طریقی به پردازنده مرکزی ربات منتقل شوند. حسگر کینکت از طریق یک کابل یو اس بی به لپ تاپ متصل می‌شود. از آن سو آردوینو به عنوان برد کنترلی ربات برای دریافت دستورات حرکتی به وسیله کابل یو اس بی به لپ تاپ متصل می‌گردد. اما برای ارسال دستورات کنترلی از برنامه به آردوینو نیاز است که پورت سریال مورد نظر برای این کار فعال باشد. دستورات کنترلی باید دقیقاً روی همان پورتی ارسال شوند که آردوینو به آن متصل است. برای این کار کتابخانه سریال پورت<sup>4</sup> به برنامه

1- Area  
2- Moment  
3- Intensity  
4- Serial port

سنسوری به عنوان ورودی در نظر گرفته می‌شود. این مسأله و نحوه قرارگیری این سنسورها باعث می‌شود تعداد قوانین زیاد نشود. سنسور 1 در جلوی ربات، سنسور 2 و 4 در سمت چپ، سنسور 3 و 5 هم در سمت راست ربات نصب شده‌اند. برای ورودی این سنسورها سه بازه تعریف شده است. این بازه‌ها بر مبنای سانتیمتر هستند. همانطور که در شکل 6 مشاهده می‌شود، اگر مانع در بازه صفر تا 5 سانتی متری ربات بود سنسور در حالت ندیدن قرار دارد. در این حالت سنسور کور می‌باشد و این بدین معنا است که مانع به ربات خیلی نزدیک است. اگر مانع در فاصله بین 5 تا 20 سانتی متر باشد، یعنی سنسور توانسته مانع را تشخیص دهد. در این حالت اگر سنسور 1 مانع را شناسایی کند بر اساس حالت دیگر سنسورها ربات به چپ یا راست حرکت می‌کند. اگر فاصله مانع بیشتر از 20 سانتی متر باشد بدین معنا است که مانع از ربات دور است و ربات می‌تواند به حرکت خود ادامه دهد.

نمودارهای دیگر سنسورها هم همانند سنسور 1 می‌باشد. طبق قوانینی که برای این بخش نوشته شده است، پس از بررسی سنسور 1 و شناسایی مانع، دیگر سنسورها هم چک شده و در صورت عدم شناسایی مانع، ربات به سمت راست حرکت می‌کند. اما اگر سنسور 1 مانع را شناسایی کرد ولی هم‌زمان سنسورهای 3 و 5 هم مانع را دیدند ربات به سمت چپ حرکت خواهد کرد [19].

سنسور های 2 و 4 در یک سمت و سنسورهای 3 و 5 در سمت دیگر ربات قرار دارند. اگر ربات به سمت راست رفته باشد، باید سنسورهای 2 و 4 مانع را شناسایی کنند. تا زمانی که این دو سنسور مانع را ببینند ربات به حرکت مستقیم خود با سرعت متوسط ادامه می‌دهد. اما زمانی که این دو سنسور مانع را نبینند، ربات به سمت چپ برگشته و به مسیر مستقیم خود با سرعت بالا ادامه می‌دهد. اگر ربات به سمت چپ رفته باشد، سنسورهای 3 و 5 مانع را شناسایی می‌کنند. به همان شکل تا زمانی که سنسورهای 3 و 5 مانع را ببینند، ربات با سرعت متوسط رو به جلو حرکت می‌کند. به محض اینکه این دو سنسور مانع را نبینند ربات به سمت راست رفته و به حرکت خود با سرعت بالا ادامه می‌دهد. برای زمانی که ربات در بین موانع از سه طرف درگیر است، به طوری که فاصله موانع با ربات کمتر از 5 سانتی متر باشد، دستور ایست به موتورها ارسال می‌گردد. در کل برای بخش فازی سازی روابط ساده در نظر گرفته شده است تا تعداد قوانین فازی زیاد نشود. در شکل 8 نمودار سه بعدی سنسور 1 و سنسور 2 به همراه موتور سمت راست دیده می‌شود. در شکل 9 نمودار سه بعدی سنسور 1 و سنسور 2 به همراه موتور سمت چپ دیده می‌شود. الگوریتم 3 شبه کد مربوط به متد فازی را نشان می‌دهد.

## 2-6- غیر فازی سازی

ماژول غیر فازی ساز رابطی است بین کنترل مبتنی بر قواعد ذکر شده و پروسه فیزیکی تحت کنترل، که به عنوان نگاشتی از فضای قوانین و متغیرهای زمانی به متغیرهای پیچیده قابل دریافت توسط پروسه عمل می‌کند و با تبدیل این متغیرها آن‌ها را به پروسه فیزیکی جهت کنترل ارسال می‌کند [6]. در بخش غیر فازی ساز به عنوان خروجی، ولتاژ موتورها مد نظر است. بازه تغییرات ولتاژ موتورها بین 0 تا 12 ولت تغییر می‌کند. اگر ولتاژ موتور بین 0 تا 6 ولت باشد، موتور در حالت توقف قرار دارد. اگر این ولتاژ بین 6 تا 9 ولت باشد، موتور با سرعت کمتری حرکت خواهد کرد. در این بازه موتور شبیه به حالتی که از پهنای عرض پالس<sup>1</sup> استفاده شده، با سرعتی کمتر

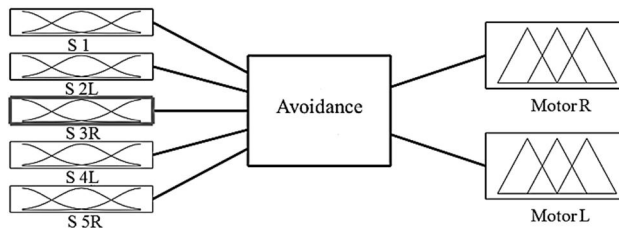


Fig. 5 Fuzzy system block with 5 input and 2 output

شکل 5 بلوک سیستم فازی به همراه 5 ورودی و 2 خروجی

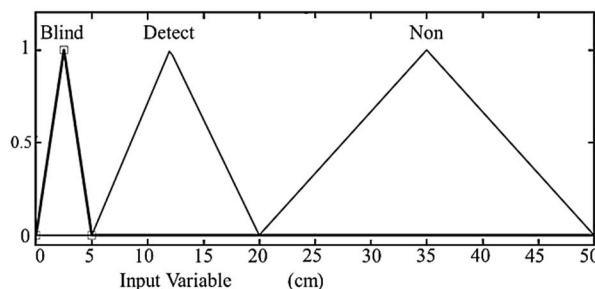


Fig. 6 Input fuzzy system from ultrasonic sensor 1

شکل 6 ورودی سیستم فازی بر مبنای سنسور مافوق صوت 1

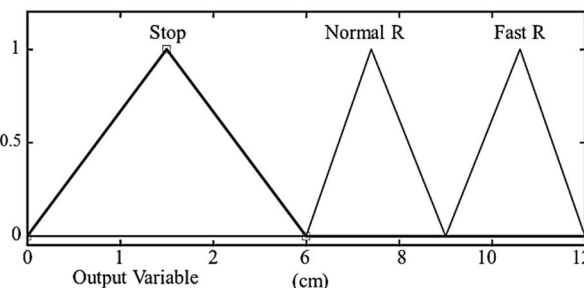


Fig. 7 Output fuzzy system from motor right voltage

شکل 7 خروجی سیستم فازی بر مبنای ولتاژ موتور

درجه عضویت (میزان تعلق این ورودی به این رنج تغییرات) این مقادیر در بازه تغییرات را نشان می‌دهد، تبدیل می‌کند. موتور استنتاج کنترلر فازی نقش کلیدی در عملکرد آن دارد که توسط یک سری قوانین اگر- آنگاه عمل می‌کند.

هر تعداد قاعده، همان تعداد خروجی کنترلر را تولید می‌کند که البته ممکن است با هم تداخل هم داشته باشند و در بعضی موارد یک قاعده 2 فرمان کنترلی نتیجه می‌دهد. اساس طراحی این قوانین پایه به تجربه طراح بسیار وابسته است که لازمه آن دانش کافی در مورد پروسه، آنالیز و مهارت- های طراحی می‌باشد. یک طراحی خوب باعث می‌شود که کنترلر کار کند و یک طراحی بهتر باعث می‌شود که با بازدهی بیشتری کار کند.

همانطور که دیده می‌شود، این سیستم 5 ورودی و 2 خروجی دارد. ورودی‌های این سیستم اطلاعات ارسالی سنسورهای مافوق صوت می‌باشند. خروجی‌های سیستم هم سطح ولتاژ موتورهای ربات می‌باشند. در این سیستم فازی تعداد 42 قانون برای رسیدن به هدف مورد نظر استفاده شده است.

یکی از نقاط ضعف در روش کنترل فازی، کند شدن سیستم بر اثر قوانین زیاد است. در صورتی که ورودی‌های سیستم زیاد باشند، قوانین به شدت افزایش پیدا کرده و محاسبه خروجی بلادرنگ نخواهد بود [18]. بنابراین کاهش ورودی‌های سیستم از اهمیت زیادی برخوردار است. به همین دلیل در این ربات از 5 عدد سنسور مافوق صوت استفاده شده که هر داده

**Algorithm 3** The pseudo-code for fuzzy logic.

```

Input: distance from ultrasonic
2: Output: voltage of motor
   if (sens1 is non)and(sens2L ← non)and(sens3R ←
   non)and(sens4L ← non)and(sens5R ← non) then
4:   motor right ← fast
   motor left ← fast
6:   if (sens1 is detect)and(sens2L ← non)and(sens3R ←
   non)and(sens4L ← non) and(sens5R ← non then
8:     motor right ← stop
     motor left ← normal
     if (sens1 is detect)and(sens2L ←
     non)and(sens3R ← detect)and(sens4L ←
     non)and(sens5R ← detect) then
10:      motor right ← normal
      motor left ← stop
12:     if (sens1 is non)and(sens2L ←
     detect)and(sens3R ← non)and(sens4L ←
     detect)and(sens5R ← non) then
14:       motor right ← fast
       motor left ← fast
       if (sens1 is non)and(sens2L ←
       non)and(sens3R ← non)and(sens4L ←
       non)and(sens5R ← non) then
16:         motor right ← normal
         motor left ← stop
18:       if (sens1 is non)and(sens2L ←
       non)and(sens3R ← detect)and(sens4L ←
       non)and(sens5R ← detect) then
20:         motor right ← fast
         motor left ← fast
         if (sens1 ← non)and(sens2L ←
         non)and(sens3R ← non)and(sens4L ←
         non)and(sens5R ← non) then
22:           motor right ← stop
           motor left ← normal
24:         if (sens1 ←
         blind)and(sens2L ← blind)and(sens3R ←
         blind)and(sens4L ← blind)and(sens5R ← blind)
         then
26:           motor right ← stop
           motor left ← stop
28:         end if
29:       end if
30:     end if
31:   end if
32: end if
33: end if
34: end if

```

**Algorithm. 3** The pseudo-code for fuzzy logic

الگوریتم 3 شبه کد برای سیستم فازی

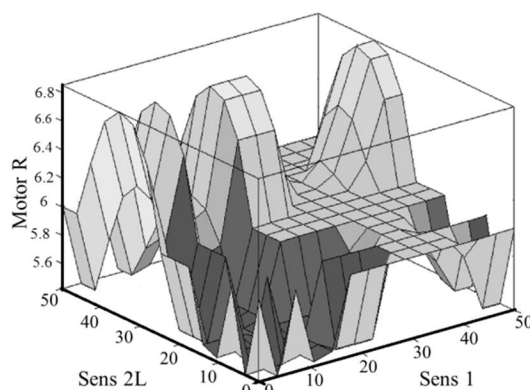
شده تا ربات کوتاهترین مسیر را بییماید. همانطور که در شکل 10 و 11 دیده می‌شود، ربات پس از شناسایی موانع شروع به چرخش کرده و موانع را دور می‌زند. پس از عبور از موانع ربات کوتاهترین مسیر را به سمت هدف می‌پیماید [22]. تست عملیاتی برنامه عدم برخورد با موانع بر روی ربات در لینک<sup>3</sup> موجود می‌باشد. همان‌طور که در فیلم مشاهده می‌شود، از انسان به عنوان موانع استفاده شده است و ربات با تشخیص موانع مسیر حرکت خود را برای عبور از موانع تغییر می‌دهد.

حرکت می‌کند. این حرکت آهسته تر باعث می‌شود که ربات در چرخش‌ها نرم‌تر حرکت کند. برای حرکت موتور با تمام سرعت هم باید ولتاژ بین 9 تا 12 ولت باشد [20].

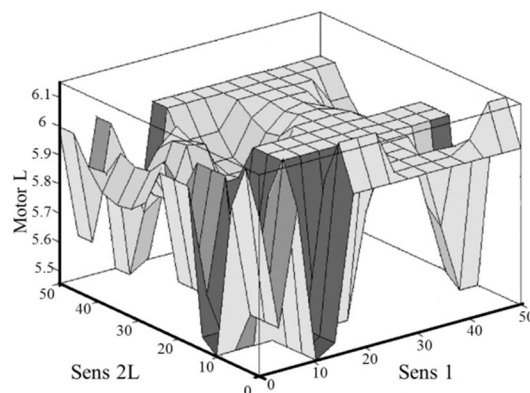
همان‌طور که در شکل 8 هم مشخص است، در بین فواصل کمتر از 10 سانتی متر سرعت موتور پایین است. به همین صورت در فواصل بالای 10 سرعت موتور بیشتر می‌شود. در شکل 9 هم در فواصل پایین سرعت موتور کم و در فواصل بالا سرعت موتور بیشتر است. البته توجه شود که این اشکال با توجه به تمام 42 قانون به دست آمده‌اند. اگر مانعی بر سر راه ربات دیده شود، برای چرخش ربات به سمت چپ یا راست از حالت متوسط<sup>1</sup> استفاده می‌شود. اگر حرکت به سمت راست باشد، یعنی موتور سمت راست در حالت توقف و موتور سمت چپ با سرعت متوسط<sup>2</sup> حرکت کند. برای اجرای این سیستم فازی تعداد 42 قانون نوشته شده است که در جدول 1 قابل مشاهده می‌باشد.

به همین صورت اگر ربات به سمت چپ بچرخد، موتور سمت چپ توقف و موتور سمت راست با سرعت متوسط می‌چرخد [21].

در مواردی که ربات مانع را شناسایی کرده و چرخش را انجام داده است، برای عبور از موانع موتورها با تمام سرعت حرکت می‌کنند. در شکل 10 نمودار حرکت ربات در مسیر با یک مانع مشخص شده است. در شکل 11 مسیر حرکت ربات با 2 مانع نشان داده شده است. از نقطه ابتدا تا نقطه هدف سعی

**Fig. 8** Surface fuzzy logic system for sensor1 and sensor2 for motor right

شکل 8 نمای سه بعدی خروجی سنسور 1 و 2 بر روی موتور سمت راست

**Fig. 9** Surface fuzzy logic system for sensor1 and sensor2 for motor left

شکل 9 نمای سه بعدی خروجی سنسور 1 و 2 بر روی موتور سمت چپ

1- Normal  
2- Normal\_L3- <http://taarlab.com/files/ravari.mp4>



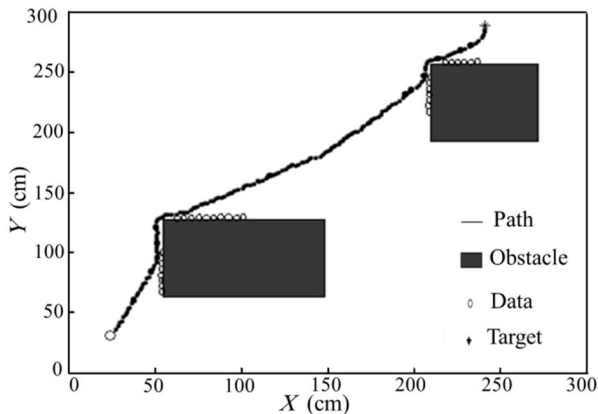


Fig. 11 The path generated by TL-PR robot when two obstacles is detected

شکل 11 ایجاد مسیر حرکت ربات TL-PR با شناسایی دو مانع

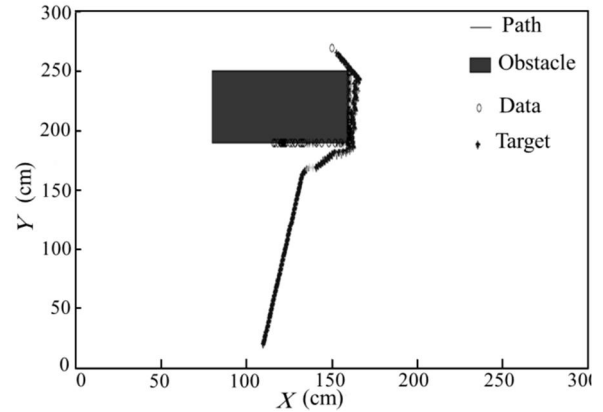


Fig. 10 The path generated by TL-PR robot when one obstacle is detected

شکل 10 ایجاد مسیر حرکت ربات TL-PR با شناسایی یک مانع

جدول 1 قوانین استفاده شده در سیستم فازی

Table. 1 Fuzzy rules used in fuzzy algorithm

نوع ارتباط	ورودی های تابع عضویت					خروجی های تابع عضویت	
	Sens1	Sens2 <sub>L</sub>	Sens3 <sub>R</sub>	Sens4 <sub>L</sub>	Sens5 <sub>R</sub>	Motor <sub>R</sub>	Motor <sub>L</sub>
AND	non	non	non	non	non	fast	fast
	detect	non	non	non	non	normal	stop
	non	non	non	non	detect	fast	fast
	detect	detect	non	detect	non	stop	normal
	non	non	non	detect	non	fast	fast
	detect	detect	non	non	non	normal	stop
	non	non	non	non	detect	fast	fast
	detect	non	detect	non	non	stop	normal
	non	non	non	detect	non	fast	fast
	detect	detect	detect	non	non	normal	stop
	non	non	non	non	detect	fast	fast
	detect	detect	detect	detect	non	stop	normal
	non	detect	detect	detect	detect	normal	normal
	non	non	non	non	detect	fast	fast
	detect	non	detect	detect	detect	normal	stop
	non	detect	non	non	detect	normal	stop
	detect	detect	non	detect	detect	stop	normal
	non	non	detect	detect	non	fast	fast
	non	non	non	blind	non	stop	normal
	non	non	non	non	non	normal	normal
non	blind	non	non	non	stop	normal	
non	non	blind	non	non	normal	stop	
non	non	non	non	non	normal	normal	
non	non	non	blind	non	stop	normal	
non	non	non	non	non	normal	normal	
non	non	non	non	blind	normal	stop	
blind	non	non	non	non	stop	normal	
blind	non	detect	non	detect	normal	stop	
blind	blind	blind	blind	blind	Stop	stop	
detect	detect	non	detect	non	Stop	normal	
non	non	non	detect	non	Fast	fast	
detect	non	non	detect	non	normal	stop	
non	detect	non	detect	non	Stop	normal	
non	non	detect	non	detect	normal	stop	
detect	detect	detect	non	non	normal	stop	
detect	non	non	detect	detect	normal	stop	
detect	blind	non	blind	non	Stop	normal	
detect	non	blind	non	blind	normal	stop	
blind	blind	blind	blind	blind	Stop	stop	
detect	blind	detect	blind	detect	Stop	normal	
detect	detect	blind	detect	blind	normal	stop	
blind	detect	detect	detect	detect	Stop	stop	

برای حرکت بهتر ربات سعی بر آن بوده که از هر دو حالت "و"، "یا" استفاده شود. با استفاده همزمان این حالت نمودارهای سه بعدی خروجی به دست آمده منطقی تر به نظر می‌آیند.

## 7- نتیجه گیری

در این مقاله بر روی طراحی و ساخت ربات TL-PR بحث شده است. ربات از نظر ساختار مکانیک مورد بررسی قرار گرفته است. یک مکانیزم ابتکاری و

ساده اما کاربردی برای ساخت ربات استفاده شده است. در بخش الکترونیک ربات سخت افزار آردوینو به همراه مازول درایور و موتورها تست شده است. در این مقاله، بر روی این ربات دو روش مجزا برای جلوگیری از برخورد ربات با موانع پیاده‌سازی شده است. روش اول برای عدم برخورد با موانع کاملاً بر اساس اطلاعاتی بوده که از طریق پردازش تصویر و کینکت به ربات ارسال شده است. در روش پردازش تصویر اگر دو مانع بر سر راه ربات قرار بگیرند، ربات آن مانعی را تشخیص خواهد داد که حجم بیشتری از فضای مقابل دوربین را اشغال نموده است. اما اگر مانعی که در مسیر ربات وجود دارند از نظر ابعادی نزدیک به هم باشند، روش پردازش تصویر باید بسیار دقیق به محاسبه حجم موانع بپردازد که این مسأله نیاز به یک الگوریتم بسیار هوشمند و سریع دارد. روش دوم بر مبنای سیستم فازی پیاده‌سازی شده است. سنسورهای فراصوتی که بر روی ربات وجود دارند، در پایین‌ترین قسمت ربات نصب شده‌اند. با استفاده از سنسورهای فراصوت مانعی که بر سر راه ربات قرار دارند با دقت مناسبی تشخیص داده می‌شوند. در نهایت اگر این دو روش باهم مقایسه گردند، پیاده‌سازی روش پردازش تصویر عملکرد بهتری بر روی ربات داشته است. البته لازم به ذکر است بهترین حالت زمانی اتفاق می‌افتد که از هر دوی این روش‌ها به صورت همزمان و از ترکیب اطلاعات سنسورها بهره برده شود. از نوآوری‌های این مقاله می‌توان به پیاده‌سازی دو الگوریتم مناسب و کاملاً متفاوت برای سیستم عدم برخورد با موانع، امکان استفاده جدا یا همزمان الگوریتم‌های مطرح شده، ساخت‌افزار قوی برای اجرای پیشرفته‌ترین الگوریتم‌های مسیریابی و پردازش تصویر بر روی ربات مذکور اشاره کرد. در آینده قصد بر آن است که روش‌های تعقیب افراد مبتنی بر ویژگی‌های هر فرد بر روی ربات پیاده‌سازی شود. همچنین روش‌هایی برای ترکیب اطلاعات سنسورها برای عملکرد هرچه بهتر ربات در آینده انجام خواهد شد. همچنین قصد بر آن است برای بهبود پایداری حرکت ربات از کنترلرهای پیشین<sup>1</sup> استفاده شود.

## 8- مراجع

- [1] X. Li, B.-J. Choi, Design of obstacle avoidance system for mobile robot using fuzzy logic systems, *International Journal of Smart Home*, Vol. 7, No. 3, pp. 321-328, 2013.
- [2] D. Itô, *Robot Vision: Strategies, Algorithms and Motion Planning*, pp. 123-170, New York: Nova Science Publishers, 2009.

1- Model predictive controller (MPC)

- Springer, 2011.
- [14] J. Borenstein, H. Everett, L. Feng, Where am I? Sensors and methods for mobile robot positioning, *University of Michigan*, Vol. 119, No. 120, pp. 27, 1996.
- [15] E. R. Melgar, C. C. Diez, *Arduino and Kinect Projects: Design, Build, Blow Their Minds*, pp. 23-99, New York: Springer, 2012.
- [16] J. R. Parker, *Algorithms for image processing and computer vision*, pp. 21-85, Indianapolis: Wiley, 2010.
- [17] A. Hornberg, *Handbook of machine vision*, pp. 35-73, Germany: Wiley, 2007.
- [18] L.-X. Wang, *Adaptive fuzzy systems and control: design and stability analysis*, pp. 37-85, New Jersey: Prentice-Hall, 1994.
- [19] M.H. Korayem, M. Nazemizadeh, H. Ghaffarpour, Optimal path planning of nonholonomic mobile robots using optimal control method and verification of the method via experimental tests of the scout mobile robot, *Modares Mechanical Engineering*, Vol. 12, No. 2, pp. 87-94, 2012. (in Persian فارسی)
- [20] T. Samu, N. Kelkar, E. L. Hall, Fuzzy Logic Control System for Three Dimensional Line Following for a Mobile Robot, *Center for Robotics Research*, Vol. 1050, pp. 72, 1996.
- [21] A. Farivar, M. Zakerzadeh, Design, Construction and Control of Two-Wheel Self-Balancing Robot, *Modares Mechanical Engineering*, Vol. 15, No. 7, pp. 188-198, 2015. (in Persian فارسی)
- [22] B. Miripour Fard, A. Hassani, S. Farhani, M. Najari, Design and fabrication of minesweeper spider robot based on the Klann mechanism, *Modares Mechanical Engineering*, Vol. 15, No. 4, pp. 321-330, 2015. (in Persian فارسی)
- [3] J. Borenstein, Y. Koren, Obstacle avoidance with ultrasonic sensors, *IEEE Journal of Robotics and Automation*, Vol. 4, No. 2, pp. 213-218, 1988.
- [4] J. Borenstein, Y. Koren, Real-time obstacle avoidance for fast mobile robots in cluttered environments, *Proceeding of Robotics and Automation Conference*, Cincinnati, IEEE, pp. 572-577, 1990.
- [5] I. Kostavelis, L. Nalpantidis, A. Gasteratos, Real-time algorithm for obstacle avoidance using a stereoscopic camera, *Proceeding of Third Panhellenic Scientific Student Conference*, Greece, 2009.
- [6] M. H. Araghi, M. Menhaj, A Combined Collision Cone and Fuzzy Based Obstacle Avoidance Algorithm for the Small Size Soccer Robots, *Proceeding of ARCS Conference*, pp. 151-158, 2008.
- [7] J. Weickert, *Anisotropic diffusion in image processing*, Teubner Stuttgart, 1998.
- [8] B. Siciliano, O. Khatib, F. Groen, *Springer Tracts in Advanced Robotics*, pp. 72-98, Berlin: Springer, 2002.
- [9] G. Bradski, A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*, pp. 16-107, California: O'Reilly Media, 2008.
- [10] S. Shukla, M. Tiwari, Fuzzy logic of speed and steering control system for three dimensional line following of an autonomous vehicle, *International Journal of Computer Science and Information Security*, Vol. 7, No. 3, USA, IEEE, 2010.
- [11] S. Hirose, A. Morishima, Design and control of a mobile robot with an articulated body, *The International Journal of Robotics Research*, Vol. 9, No. 2, pp. 99-114, 1990.
- [12] M. Banzi, *Getting started with Arduino*, pp. 17-68, California: O'Reilly Media, 2011.
- [13] J.-D. Warren, J. Adams, H. Molle, *Arduino for Robotics*, pp. 51-82,