

اجتناب از موانع فازی برای روبات های سیار همه جهته

محمد نخبه زعیم و دانیال خشابی

{d.khashabi,nokhbeh100}@gmail.com

دانشکده مهندسی برق، دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

پروژه ی پایانی درس هوش محاسباتی

چکیده: در این پروژه ابتدا یک محیط شبیه سازی برای بررسی حرکت یک روبات همه جهته دارای سنسورهای حاشیه ای در نرم افزار MATLAB انجام شد. روبات مورد نظر از یک نقطه ی ابتدایی شروع به حرکت می کند تا اینکه بعد از عبور از تعدادی از موانع تصادفی به نقطه ی انتهایی حرکت خود برسد. در این پروژه این مساله در لایه های مختلفی برخورد شده است؛ ابتدا سعی شده است مساله تنها با روش فازی با استراتژی فازی حل شود. در ادامه برای بهینه سازی جواب ها الگوریتم فازی با برخی از الگوریتم های مسیریابی ادغام شده است.

۱. مقدمه

آموزش روبات برای اجتناب از موانع و رسیدن به نقطه ی هدف از جذابترین مباحث در روباتیک است. تاکنون روش های متنوعی برای حرکت روبات برای اجتناب از موانع ارائه شده است. یکی از روش های متداول روش میدان های پتانسیلی مصنوعی است یا مدل نیروهای مجازی است [۵،۴]. چنین روش تاکنون بسیار پیشرفت کرده و کارامدی بسیاری در هدایت نرم و روان روبات ها از کنار موانع دارد. اگر در این مدل های ساده و کارآمد نیز می توان حالت هایی را پیدا کرد که روبات به دام بیافتد و حرکت نکند. همچنین نیروی دافعه از سوی هر جسم نزدیک به روبات نیز می تواند مانع از حرکت آن از میان موانع با فاصله ی بسیار کم شود. در شرایطی نیز ممکن است روبات بین دو مانع شروع به حرکت نوسانی کند. در [۶] سعی شده است با پیش بینی وضعیت آینده و با استفاده از روش فازی حرکت روبات انجام شود. عملکرد مطلوب الگوریتم برای حالت هایی نشان داده شده است. در [۷] نیز سعی شده است که پارامترهای مدل فازی برای حرکت در محیط توسط الگوریتم ژنتیک بهینه شود. روش های بسیاری دیگری نیز وجود دارند که در اینجا از آوردن آنها خودداری می کنیم. علاقمندان به مطالعه می توانند به [۵] مراجعه کنند.

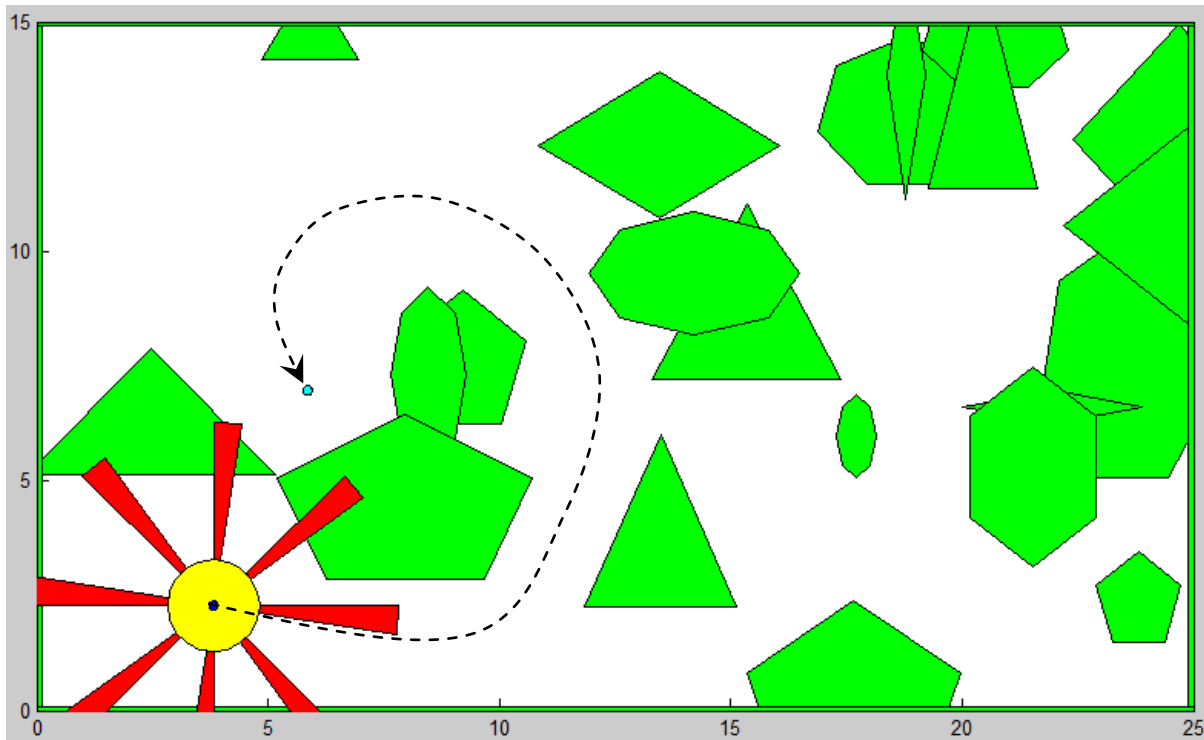
در ادامه کارهای انجام شده در این پروژه به طور خلاصه بیان می شود. در قسمت اول ساختار محیط شبیه سازی در MATLAB معرفی می شود. در ادامه مدل فازی برای حرکت روبات؛ و در قسمت نهایی مدل ترکیبی با استفاده از الگوریتم A^* و مدل فازی ارائه می گردد. در انتها نیز برخی از عملکردهای روبات آورده شده است. لازم به ذکر است که کدهای شبیه سازی این پروژه به صورت آزاد برای دسترس عموم قرار داده شده اند^۱.

۲. ساختار محیط و روبات

۲.۱. ساختار محیط

محیط شبیه سازی، شامل یک روبات متحرک در یک محیط با تعدادی مانع چند ضلعی تصادفی است. روبات باید از نقطه ای تصادفی در این محیط شروع کرده، بدون برخورد به موانع به نقطه ی نهایی برسد. نمونه ای از شکل محیط مورد آزمایش در شکل ۱ آورده شده است. لازم به ذکر است که روبات قبل از شروع به حرکت از محیط هیچ ایده ای در مورد طرز قرار گیری اجسام در آن ندارد. در واقع تنها اطلاعات روبات از محیط توسط تعداد محدودی سنسور در اطراف آن تامین می شود که خود آنها نیز دارای بُرد محدود هستند.

¹ <http://ele.aut.ac.ir/~khashabi/fuzzyAStarMotion.html>



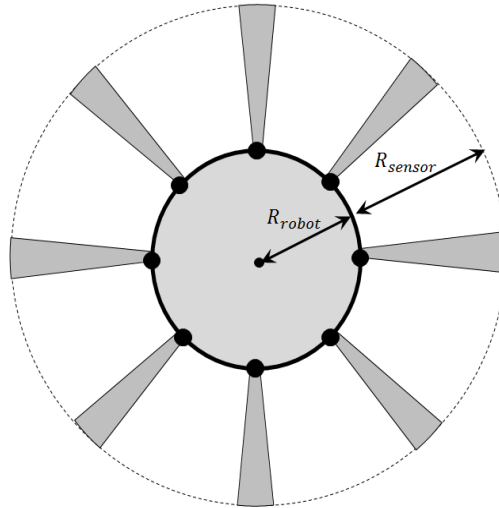
شکل ۱: شکلی از محیط تصادفی موانع در محیط؛ موانع چند ضلعی به صورتی تصادفی چیده شده اند. همچنین محیط دارای یک نقطه ی شروع و پایان است.

۲.۲. ساختار روبات ۱: روبات همه جهته

ساختار شکل این روبات در شکل ۲ نمایش داده شده است. این روبات همه جهته^۲ نامیده می شود؛ زیرا قابلیت این را دارد در همه ی جهت ها بدون نیاز به چرخش حرکت کند. در شکل نمایش داده شده، روبات مورد نظر دارای ۸ سنسور ultrasonic برای تشخیص فاصله ی اجسام از خود است. این سنسور ها قابلیت تشخیص اجسام تا فاصله ی R_{sensor} از خود هستند. معمولاً در آزمایش های انجام شده مقادیر شعاع روبات و شعاع سنسورهای فاصله به ترتیب $R_{sensor} = 1m$ و $R_{sensor} = 3m$ در نظر گرفته می شود. همچنین روبات در هر لحظه مکان خود و مکان نقطه ی انتها را در اختیار دارد.

برای آزمایش های اولیه فرض می شود که روبات قادر است با سرعت ثابتی حرکت کند. در این حالت تصمیم گیری برای حرکت نسبتاً آسان است. معمولاً مقدار سرعت در آزمایش ها $V_{max} = 0.05 \frac{m}{s}$ در نظر گرفته می شود.

²Omnidirectional

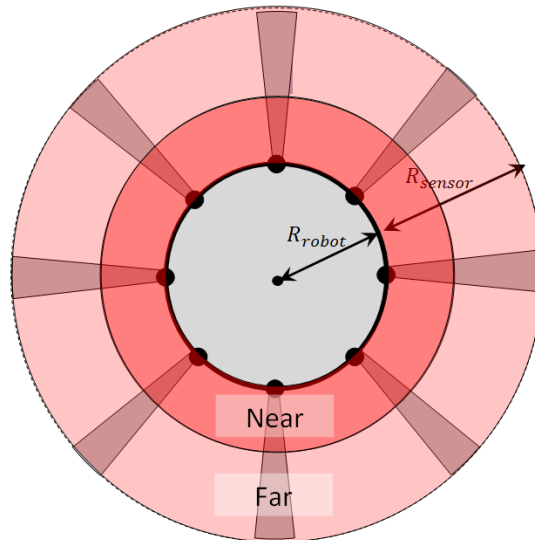


شکل ۲: نمایش ساختار کلی روبات تمام جهته، دارای ۸ سنسور.

برای مدلی واقعی تر روبات دارای لختی در نظر گرفته می شود؛ یعنی روبات می تواند با شتاب ثابت حرکت کند تا اینکه به سرعت حداکثر برسد. در این حالت شتاب حداکثر و سرعت حداکثر به ترتیب $v_{max} = 0.05 \frac{m}{s}$ و $a_{max} = 0.01 \frac{m}{s^2}$ در نظر گرفته شدند.

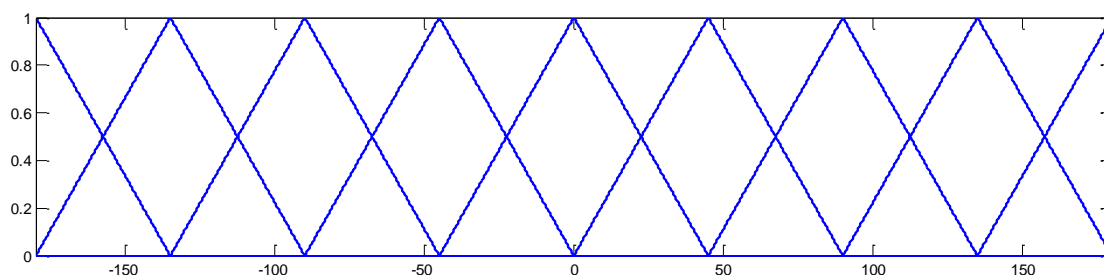
۳. روش فازی برای اجتناب از موانع

در تصمیم گیری فازی لازم است براساس موقعیت موانع نسبت به روبات تصمیم گیری کنیم. ناحیه بندی فضای فاصله ی اجسام از روبات به صورت شکل ۳ نشان داده شده اند. معمولا فاصله ی $R_{Near} = 1m$ و $R_{Far} = 2m$ در نظر گرفته می شود.



شکل ۳: ناحیه بندی فاصل از سنسورها برای تصمیم گیری فازی.

همچنین حرکت روبات در راستاهای مختلف در شکل ۴ نشان داده شده است. در ارائه ی الگوهای حرکت فازی از [۴,۳,۲] الگو گرفته شده است.



شکل ۴: ناحیه بندی فازی حرکت روبات در جهت های مختلف.

۳.۱. استراتژی حریصانه

در استراتژی حریصانه^۳ سعی بر این است که در سریع ترین حالت ممکن به هدف مورد نظر برسیم؛ مگر آنکه به مانعی برسیم. ساختار قوانین برای تصمیم گیری حریصانه به صورت زیر است:

IF $d(\text{object}, \text{sensor}_i) < R_{\text{near}}$ TEHN
move in oposite i direction

در حالت عادی که هیچ جسم خارجی در ناحیه ی Near سنسور ها نیست، رفتار با توجه به مکان نسبی نقطه ی هدف انجام می گیرد:

IF $\forall i \ d(\text{object}, \text{sensor}_i) > R_{\text{near}}$ THEN
move toward the goal

با توجه به دستورات فوق می توان از قبل پیش بینی کرد که مدل فوق همواره دارای تمایل به حرکت به سمت نقطه ی هدف است، مگر آنکه در حال برخورد به مانعی باشد، که در اینصورت از آن دور می شود. در شرایطی که مسیر مورد نظر بن بست باشد، این استراتژی قادر به بازگشت و انتخاب مسیر جدید نیست. لذا روبات در بن بست به دام می افتد.

۳.۲. مدلی برای چرخش اجسام

در این مدل سعی می شود ساختار به گونه ای طراحی شود که روبات قادر باشد که اجسام را دور بزند. به مجموعه قوانین حریصانه ی قبل، قوانین دیگری اضافه می شود که در صورتی که به جسمی رسید، در صورتی که در فاصله ی Far از جسم بود به سمت راست آن حرکت کند:

IF $d(\text{object}, \text{sensor}_i) > R_{\text{far}}$ and $d(\text{object}, \text{sensor}_{i-1}) < R_{\text{far}}$ TEHN
move in $(i + 1)$ direction

محاسبه ی جهت حرکت به سمت راست با استفاده از فاصله ی جسم از سنسورهای روبرویی و تخمین شیب نسبی جسم نسبت به روبات انجام می گیرد.

نتایج آزمایش این مدل نیز نشان داد که امکان به دام افتادن روبات در این مدل نیز وجود دارد. در ضمن عملیات گردیدن حول اجسام به سمت راست آنها در بسیاری از شرایط به هیچ وجه بهینه نیست.

۳.۳. اشکالات مدل ها

مشاهده شد هرکدام از این روش ها مزایا و معایب خود را دارا هستند. اما در کل می توان گفت برای همه ی روش ها می توان حالت هایی را پیدا کرد که مدل مذکور در آن حالت به دام بیافتند. چرا که تمامی مدل های فازی فوق در توجه به شرایط و وضعیت های محلی مشترک هستند.

۴. ادغام با روش های دیگر

³Greedy

با توجه به نتایج بدست آمده از پیاده سازی های الگوریتم فازی برای اجتناب از موانع فازی در قسمت قبل، می توان گفت رسیدن به انتهای مسیر برای این مدل ها تضمین شده نیست. چرا که مدل های طراحی شده تنها بر اساس شرایط کنونی برای حرکت روبات تصمیم گیری می کنند. لذا ممکن است در گوشه هایی از نقشه در بین موانع به دام بیافتند. برای رفع این مشکلات لازم است نقشه ی بدست آمده از محیط تا لحظه ی کنونی را ذخیره کرده و از اطلاعات آن استفاده کنیم. در ادامه روش هایی را معرفی می کنیم که نشان می دهند این استفاده به چه صورت می تواند انجام گیرد.

۴.۱. الگوریتم A^* برای مسیریابی

در مقاله ی [۲] سعی شده است از برتری های اجتناب از موانع فازی همراه با الگوریتم مسیریابی A^* استفاده شود. در مقاله ی مذکور فرض شده است محیط کاملا شناخته شده است. در صورتیکه در مساله ی ما روبات به تدریج در محیط حرکت کرده و آن را می شناسید. در نتیجه مساله دشوار تر است.

الگوریتم A^* قابلیت این را دارد که مسیریابی برای رسیدن به نقطه ی انتها را در محیطی شناخته شده را انجام دهد. اما چندان دستوری را برای حرکت نرم و روان روبات در یک محیط ارائه نمی کند. در صورتی که فازی، ساختاری فراهم می آورد که روبات بتواند به صورتی نرم و روان در محیط دارای موانع حرکت کند. در عوض قابلیت این را ندارد که مسیریابی را با توجه به نقشه ی محیط اطراف و مسیر طی شده تاکنون انجام دهد و رسیدن به نقطه ی انتهایی را تضمین کند.

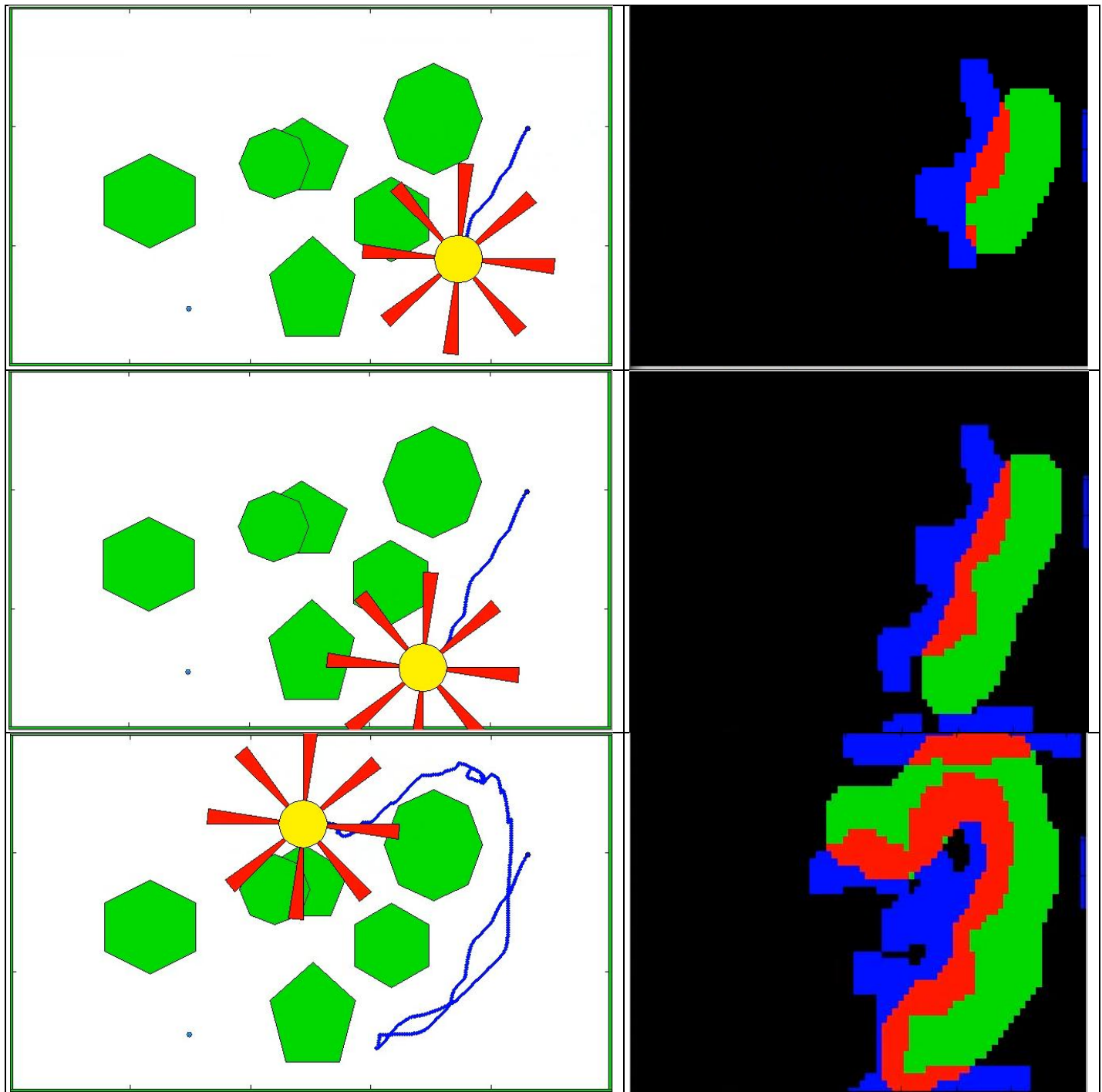
همانطور که ذکر شد الگوریتم A^* برای حرکت در محیطی شناخته شده است. در صورتی که در مساله مورد نظر در اینجا محیط از ابتدا شناخته شده نیست. اما تدریج که محیط بررسی می شود، اطلاعات بیشتری از محیط به دست می آید. به این ترتیب که در ساختار روبات نقشه ای مجازی از محیط طراحی می شود که در آنها سه ناحیه در دست است:

۱- نواحی مشاهده نشده.

۲- نواحی مشاهده شده که مانع هستند.

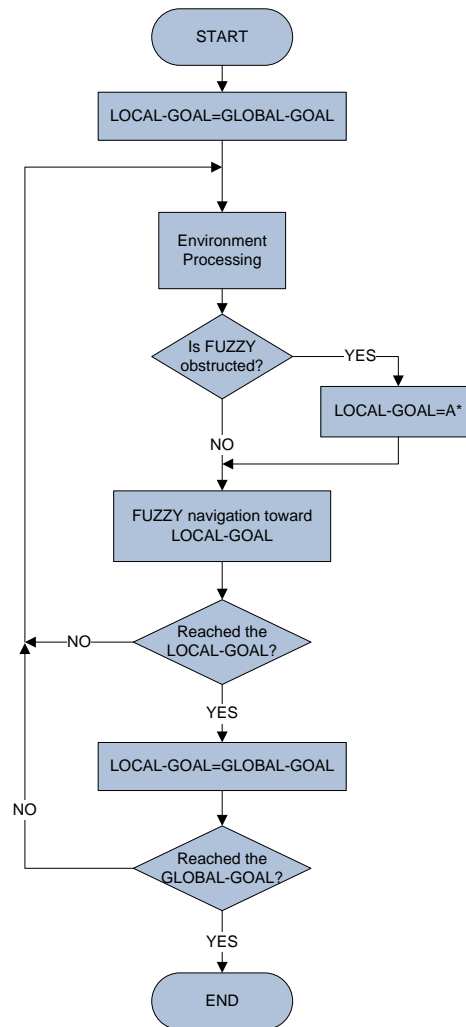
۳- نواحی مشاهده شده که مانع نیستند.

نمونه ای از چنین نقشه ی تهیه شده توسط روبات در شکل ۵ نشان داده شده است. نواحی مشاهده نشده با رنگ سیاه، نواحی مشاهده شده که مانع هستند با رنگ آبی و نواحی که مشاهده شده اند ولی مانع نیستند با رنگ سبز مشخص شده اند. برای تمییز ناحیه ی بین سبز و آبی از رنگ قرمز استفاده شده است.



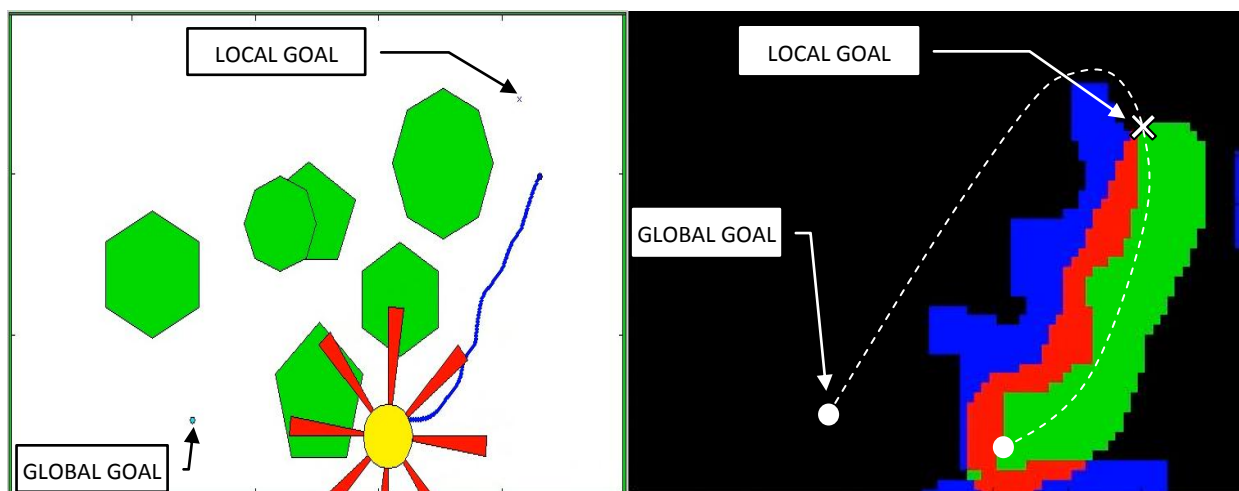
شکل ۵: نمایش تشکیل تدریجی نقشه ی محیط توسط روبات با حرکت در محیط.

الگوریتم اجرا شده در این قسمت در شکل ۶ نمایش داده شده است. روبات در شرایط عادی به دنبال GLOBAL-GOAL یا هدف نهایی است. در صورتیکه احساس کند که گیرکرده است، با استفاده از الگوریتم A^* به دنبال یک LOCAL-GOAL حرکت می کند (به جای GLOBAL-GOAL) تا اینکه در نهایت بعد از رسیدن به LOCAL-GOAL مورد نظر، دوباره شروع به رفتن به سمت GLOBAL-GOAL می کند.



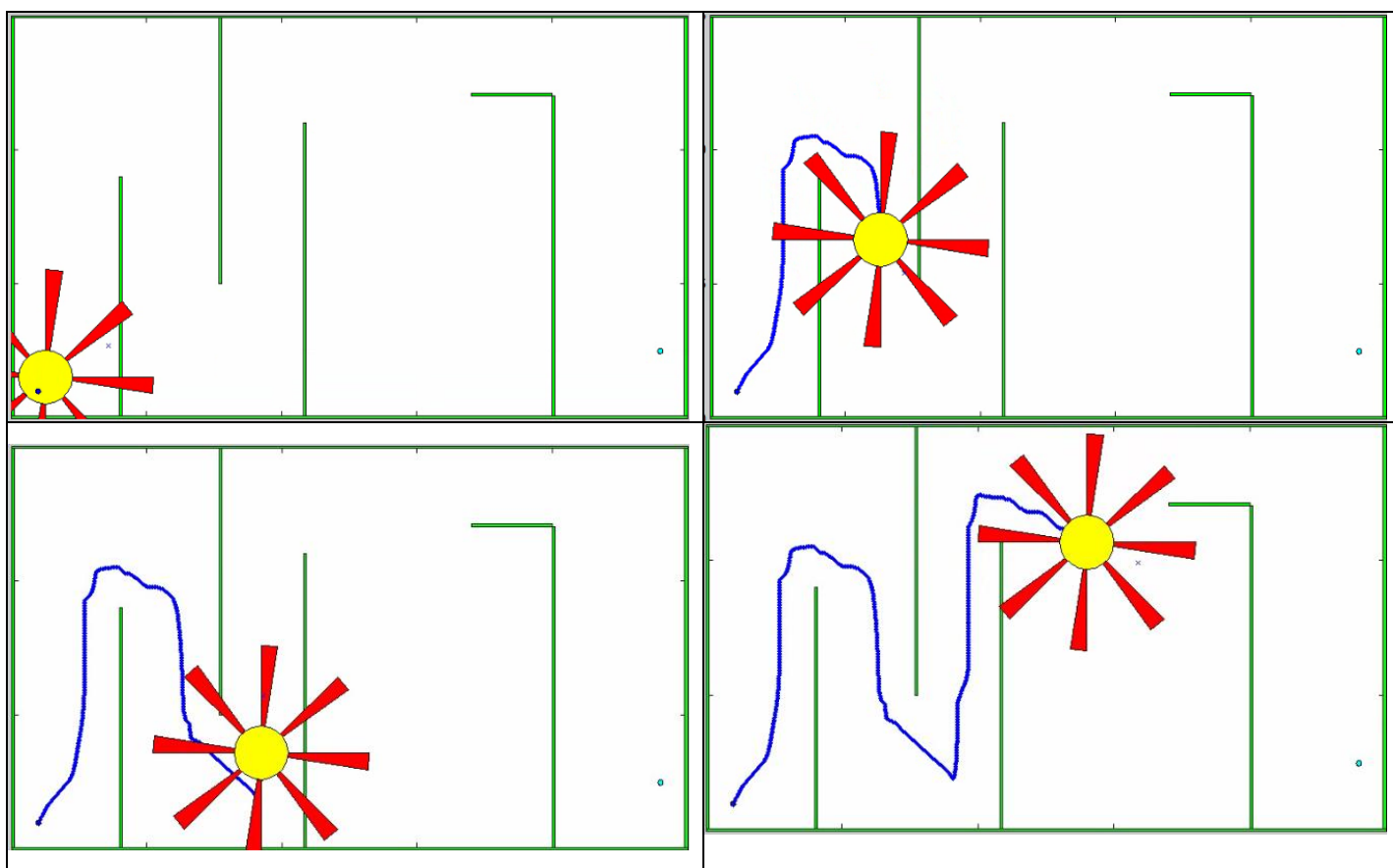
شکل ۶: نمایش فلوچارت استفاده ی ترکیبی از روش فازی به همراه مسیریابی A*

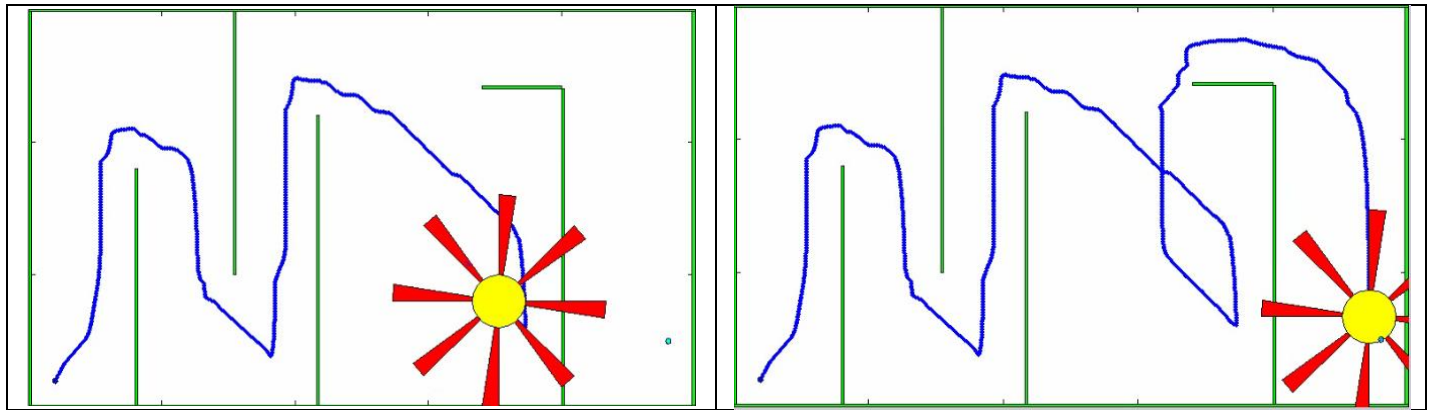
برای پیدا کردن LOCAL-GOAL الگوریتم A* تمام نقاط دیده نشده را "دیده شده" فرض می کند و نقطه ی جدید را مرز بین ناحیه ی دیده شده و دیده نشده قرار می دهد. نمونه از چنین انتخابی در شکل ۵ نمایش داده شده است. با این کار روبات شروع به حرکت فازی به سمت نقطه ی فازی می کند. در نهایت اگر فضا خالی باشد از ناحیه عبور می کند تا هدف برسد. در صورتی که مانع باشد، نقشه ذهنی ربات کامل تر شده و به دنبال مسیرهای دیگری می گردد.



شکل ۸: نمایش انتخاب یک LOCAL-GOAL توسط الگوریتم A*

نمایشی از حرکت روبات با استفاده از این روش در شکل ۹ نشان داده شده است.





شکل ۹: حرکت روبات در یک نقشه ی آزمایشی.

منابع و مراجع

- [1] LaValle, S.M., Planning algorithms, Cambridge University Press, 2006.
- [2] Gerdelan, A.P. and Reyes, N.H., A novel hybrid fuzzy A* robot navigation system for target pursuit and obstacle avoidance, Proceedings of the First Korean-New Zealand Joint Workshop on Advance of Computational Intelligence Methods and Applications, pp. 75—79, 2006.
- [3] Yen, J. and Pfluger, N., A fuzzy logic based extension to Payton and Rosenblatt's command fusion method for mobile robot navigation, IEEE Transactions on Systems, Man and Cybernetics, 1995.
- [4] McFetridge, L. and Yousef Ibrahim, M., New technique of mobile robot navigation using a hybrid adaptive fuzzy potential field approach, Elsevier Computers & industrial engineering, 1998.
- [5] Khatib, O., Real-time obstacle avoidance for manipulators and mobile robots, The international journal of robotics research, 1986.
- [6] Maeda, M. and Shimakawa, M. and Murakami, S., Predictive fuzzy control of an autonomous mobile robot with forecast learning function, Fuzzy Sets and Systems, 1995.
- [7] Joo, YH and Hwang, HS and Kim, KB and Woo, KB, Fuzzy system modeling by fuzzy partition and GA hybrid schemes, Elsevier, 1997.