

طراحی يك كنترل كننده فازي تطبيقی برای ناوبری ربات های متحرك با استفاده از يادگيري تقويتی

دانشگاه اميركبير

دانشكده مهندسی كامپيوتر

محمد رضا ميبدی

شراره بابويه

چكیده

(۱) مقدمه

در این مقاله مراحل طراحی يك كنترل كننده فازي تطبيقی، جهت رفتار ناوبری توسط دو ربات متحرك در يك محيط ناشناخته شرح داده شده است. هر ربات قادر است با كمك حسگرهای خود شرایط جاری محیط را درك كند و با كمك كنترل كننده خود تصميم مناسبی اتخاذ و در محیط اجرا نماید. ورودی كنترل كننده مقادير حسگرهای ربات و خروجی آن سرعت چرخ های چپ و راست ربات هستند. جهت تعیین پارامترهای توابع عضویت فازي ورودی و خروجی، با توجه به شرایط محیط در هر قدم زمانی از اتوماتون یادگیر استفاده می نماییم. ما جهت یافتن قوانین فازي برای ناوبری هر ربات از الگوریتم یادگیری Q استفاده می کنیم، اما برای همکاری ربات ها از الگوریتم یادگیری Q با تابع ارزش توزیع شده استفاده میشود. ربات در محیط گردش می کند و در حالت های مختلف عمل های مجاز مختلف را می آزماید و مقادير Q را بر روزر می آورد. این مقادير Q جهت یافتن قوانین فازي مناسب در هر قدم زمانی به کار میروند. هر رفتار به طور جداگانه پیاده سازی شده و سپس روشی جدید جهت ترکیب رفتارها معرفی می نماییم. در انتها با كمك آزمایش های مختلف کارایی راهبرد ارائه شده را نشان می دهیم.

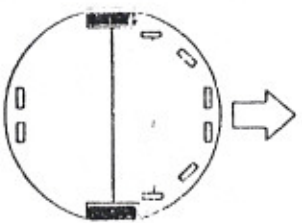
الگوریتم ما به دلیل عدم استفاده از رهبردهایی نظیر شبکه های عصبی یا الگوریتم های تکاملی که در اغلب تحقیقات مورد استفاده قرار گرفته اند، ساده و روشن میباشد و چون نیازی به مشتق گیری و یا سایر عملیات پیچیده ریاضی ندارد کم هزینه و سریع می باشد. ربات با كمك این كنترل كننده ساده میتواند مسیری نرم، هموار، کوتاه و حتی مناسب تر از روش های مذكور انتخاب کرده و به سمت هدف خود پیش رود. در این كنترل كننده ترکیب رفتارها به سادگی، با روشی قابل درك و با حداقل هزینه ممكن عملی میباشد. به کارگرفتن اتوماتون یادگیر باعث می شود كه كنترل كننده خود را با تغییرات محیط تطبیق دهد و در نتیجه با عرض شدن شرایط محیط عملکرد آن کاهش نمی یابد. نکته مهم این است كه تصميم گیری با اطلاعات بسیار ناچیز در مورد طبیعت محیط صورت می گیرد و ربات بدون داشتن هرگونه پیش فرضی به تنهایی میتواند نحوه تصميم گیری، انتخاب و اجرای عمل مناسب را بیاموزد.

ناوبری برای ربات های متحرك عبارت است از حرکت به طرف يك هدف مشخص درحالیكه ربات از برخورد با موانع پرهیزی نماید. در يك محیط دیناميك كه دائماً در حال تغییر است، استفاده از روش های سراسری غیر ممكن یا بسیار پرهزینه است، چرا كه در این روش ها مدل ریاضی و یا نقشه كامل محیط مورد نیاز می باشد. اما تهیه مدل ریاضی و یا نقشه كامل محیط کاری پیچیده است و با تغییر محیط ناگزیر از به روز در آوردن مدل ریاضی و یا نقشه آن خواهیم شد. باین ترتیب روش های سراسری برای حل چنین مسائلی غیر عملی می باشند. پس به ناچار باید از روش های محلی استفاده نمود.

در این روش های محلی ربات در محیط گردش می کند و اطلاعات مورد نیاز خود را با كمك حسگرهایش به صورت محلی جمع آوری می نماید. ربات میتواند بر اساس این اطلاعات درك شده، در حالت های مختلف محیط تصميم های مناسب اتخاذ نموده، عمل درستی را انجام دهد. به عبارت دیگر حسگرهای ربات از طریق يك پردازنده ساده مانند شبکه های عصبی یا سیستم های فازي و غیره، مستقیماً به عملگرهای ربات متصل می شوند و اطلاعات خام در هر لحظه از زمان به ایجاد عملیات موتوری منجر می گردند.

سیستم های استنتاج فازي، به خاطر معماری ساده و هزینه محاسباتی پایین، در زمینه رباتيك توجه زیادی را به خود جلب نموده اند. يك سیستم استنتاج فازي به تنهایی برای يك محیط ساده كه قوانین حاكم بر آن را به آسانی می توان استخراج نمود كافی است [۱]. ولی برای محیط های پیچیده تر استخراج و تنظیم قوانین فازي لازم می باشد. برخی از محققان برای بدست آوردن قوانین فازي از يك شبکه عصبی كه وزن های آن نموده اند [۲]، [۳]، [۴].

کیپرا از دو لایه متناظر با دو برد اصلی تشکیل شده است؛ برد حسسی حرکتی و برد CPU سیستم موتوری شامل دو چرخ جانبی و دو Pivot در جلی و عقب است. این پیکربندی برای مقابله با موانع هندسی پیچیده بسیار مناسب است چرا که باعث می شود ربات بتواند بدون تغییر مکان در جا بچرخد. کیپرا دارای نوعی حرکت است که بر عملگرهای جانبی و متقارن استوار بوده و در اغلب جانوران دیده می شود. در ضمن ساختار مکانیکی و سینماتیک کیپرا منطبق بر مدل سینماتیکی 1 از مدل های سینماتیکی معادل است. سیستم سنسوری ربات پایه بسیار ساده و از هشت سنسور مسافت یاب مادون قرمز تشکیل شده است که بر محیط ربات پخش شده اند. این پیکربندی سنسوری در شکل (۲) نشان داده شده است.



شکل (۲) پیکربندی سنسورهای مسافت یاب

این ترکیب بندی باعث حساسیت بیشتر در یک سمت ربات می گردد که می توان آنرا بعنوان چپری ربات شناسایی کرد. سنسورهای فاصله از یک فرستنده و یک گیرنده نورمادون قرمز تشکیل شده است. آنها می توانند حضور اشیاء را با فرستادن و اندازه گرفتن نور منعکس شده ردیابی کنند. از این سنسورها همچنین می توان بعنوان سنسورهای نورمادون قرمز غیر فعال استفاده کرد.

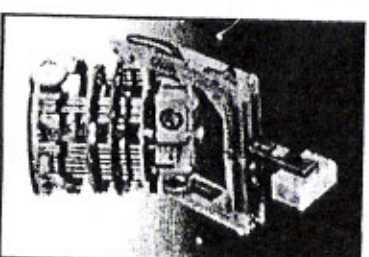
کیپرا مجهز به باتریهایی است که آنرا تا مدت زمان مشخصی خوراکها می دارند. برد CPU یک پردازنده موتورولا همراه با RAM و EEPROM را در خود جای داده است. یک مبدل A/D اجازت کسب سیگنال های آنالوگ واصله از برد حسسی - حرکتی را می دهد. یک خط سربال برای اتصال ربات به کامپیوتر و نیز تغذیه انرژی در ربات تعبیه گشته است. ارتباط الکتریکی بین برد CPU و برد حسسی - حرکتی توسط اتصالاتی که پایه مکانیکی نیز هستند، برقرار می گردد. این اتصالات یک Bus توسعه را برای اضافه کردن ماچول های دیگر به ربات فراهم می کنند.

Kubota و Fukuda از یک شبکه عصمی و یک قانون مکاشفه ای جهت بدست آوردن تالی های قوانین فازی استفاده نموده اند [۵]. در برخی از تحقیقات الگوریتم های ژنتیکی و تکامل مستقیماً برای مشخص کردن و یا بهینه نمودن قوانین فازی به کار رفته اند [۶]. در این تحقیق ما قوانین فازی را با استفاده از یادگیری Q در هر قدم زمانی انتخاب می نماییم.

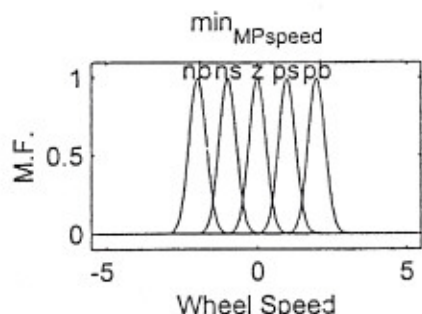
در این مقاله ابتدا به معرفی ربات کیپرا که در شبیه سازی ها مورد استفاده قرار گرفته است می پردازیم. سپس در بخش ۳ ساختار کنترل کننده را شرح می دهیم. اتوماتون تصادفی یادگیر و الگوریتم یادگیر به ترتیب در بخش های ۴ و ۵ مفصلاً توضیح داده خواهند شد. بخش ۶ به رفتار همکاری اختصاص داده شده است. در بخش ۷ به بررسی عملکرد کنترل کننده فازی تطبیقی طراحی شده می پردازیم و نتایج آزمایش ها را بیان می نماییم. بخش آخر شامل نتایج و پیشنهادات می باشد و به دنبال آن لیست تعدادی از مراجع مورد استفاده در این تحقیق ملاحظه می فرمایید.

۲) ربات کیپرا

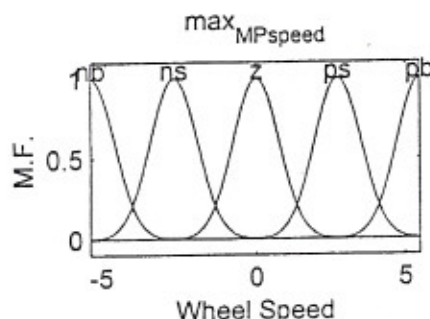
کیپرا یک ربات استوانه ای به قطر ۵۵mm و با ارتفاع متغیر است. این ربات مینیاتوری توسط A. Guignard, E. Franzی , F. Mondada در آزمایشگاه میکرو انورماتیک استیتوپولی تکنیک قدرال لوزان واقع در سوئیس، طراحی و توسعه یافته و اکنون توسط شرکت K-Team بطور تجاری تولید می گردد. اندازه کیپرا به آزمایشگر اجازه می دهد که یک محیط بزرگ را در یک سطح محدود محقق کند (شکل ۱).



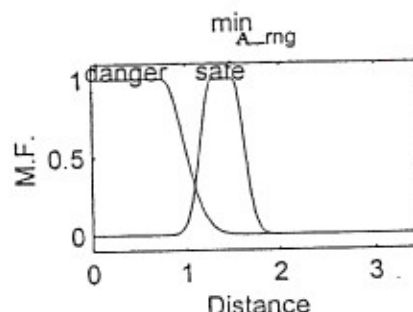
شکل ۱ ربات کیپرا



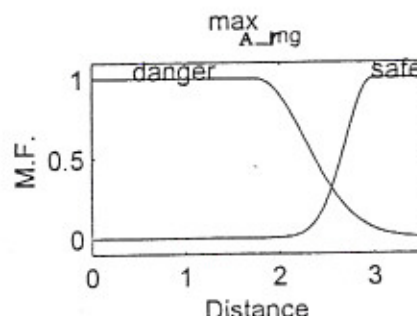
توابع عضویت فازی خروجی بازای
حداقل MP-Speed



شکل ۴- توابع عضویت فازی خروجی بازای
حداکثر MP-Speed



توابع عضویت فازی ورودی بازای
حداقل A-mg



شکل ۳- توابع عضویت فازی ورودی بازای
حداکثر A-mg

نزدیک ترین مانع را در جهت خود اندازه گیری می نماید. این مقدار با b نشان داده می شود. برای هر یک از متغیرهای ورودی و خروجی تعدادی متغیر زبانی و تابع عضویت فازی تعریف می شود. توابع عضویت فازی گاوسی شکل انتخاب شده اند. پارامترهای این توابع برای هر متغیر زبانی و ورودی بر حسب برد توجه حسگرهای ربات (A_mg) و برای هر متغیر زبانی خروجی بر حسب حداکثر سرعت مجاز ربات (MP_Speed) در هر قدم زمانی بصورت تطبیقی انتخاب می شوند. این کار با کمک اتوماتون تصادفی یادگیر خطی انجام می شود.

به شکل ۲ توجه فرمایید. این شکل توابع عضویت فازی ورودی را بازای حداقل و حداکثر برد توجه نشان می دهد. شکل ۴ توابع عضویت فازی خروجی را بازای حداقل و حداکثر (MP_Speed) یا حداکثر سرعت مجاز ربات نشان می دهد. در این شکل z (zero), b (big), s (small), p (positive), n (negative) می باشند.

مادر آزمایشات خود فرض نموده ایم که ربات حداکثر با سرعتی برابر با قطر خود حرکت می نماید و بیشترین فاصله ای که هر حسگر ربات در مقابل خود تشخیص می دهد، بر اساس میزان پراکندگی موانع در محیط بین ۸۲ تا ۱۹۲ میلی متر متغیر است. در شبیه سازی ما تمام مقادیر با ۵۵ میلی متر که قطر ربات می باشد مقیاس شده اند، و ربات تنها از حسگرهای جلو و طرفین خود برای ناوبری استفاده می نماید.

۳) ساختار کنترل کننده فازی

برای کنترل ربات از یک کنترل کننده فازی تطبیقی استفاده می نماییم. این کنترل کننده سه ورودی از سنسورها دریافت می نماید که به ترتیب، فاصله تا نزدیک ترین مانع سمت چپ، جلو و سمت راست ربات می باشند. کنترل کننده دو خروجی دارد که سرعت های چرخ چپ و چرخ راست می باشند. ربات در محیط گردش می کند، و در هر قدم زمانی هر سنسور ربات مانند فاصله ربات تا

قوانین فازی به شکل if-then بیان می شوند. مثلاً قانون ۱ام را می توان به کمک رابطه (۱) نشان داد. این قوانین در هر قدم زمانی با کمک الگوریتم یادگیری Q مشخص می شوند. در این رابطه در قسمت (۵) جزئیات بیشتری بیان شده است.

IF d_1 is $ILV_{i,1}$ and d_2 is $ILV_{i,2}$ and d_3 is $ILV_{i,3}$
THEN v_1 is $OLV_{i,1}$ and v_r is $OLV_{i,2}$ (۱)

ILV و OLV به ترتیب Input Linguistic Variable و Output Linguistic Variable میباشند.

سیستم استنتاج فازی مورد استفاده، سیستم استنتاج فازی مدانی است که بر اساس عملگرهای استنتاج \min_max عمل می کند. روش دفازی کردن Center_of_Area می باشد. شکل ۵ معماری کلی کنترل کننده را نشان می دهد. برای درک روشن تر نحوه ارتباط کنترل کننده فازی، اتوماتون یادگیر و الگوریتم یادگیری Q به شکل ۶ توجه نمایید.

۴) شرح اتوماتون یادگیر

یک اتوماتون یک ماشین یا مکانیزم کنترلی است، با این قابلیت که می تواند خود را با تغییرات محیط تطبیق دهد. یک اتوماتون بدون داشتن هرگونه اطلاعات در مورد عمل بهینه سعی بر یافتن جواب مناسب برای مساله دارد. در آغاز مقدار احتمال برای تمام عمل های یکسان است. عملی به تصادف انتخاب می گردد، سپس پاسخ محیط به این عمل خاص مشاهده می شود. مقدار احتمال این عمل بر اساس این پاسخ محیط بروز در آورده می شود و این مراحل مجدداً تکرار می شوند. یک اتوماتون که به شکل فوق سعی در بهبود عملکرد خود داشته باشد اتوماتون یادگیر گفته می شود [۸].

طراحی رفتارهای مختلف برای یک ربات متحرک در یک محیط دینامیک (دائماً در حال تغییر) کاملاً وابسته به شرایط محیط می باشد. در طراحی رفتار پرهیز از مانع، کنترل برد توجه ربات و نیز سرعت حرکت آن با توجه به تعداد موانع در عملکرد سیستم کنترل و بروز رفتار دلخواه بسیار موثر است. هرگاه تعداد موانع در اطراف ربات زیاد باشد، لازم است ربات بیشتر روی نواحی نزدیک تر به خود تمرکز داشته باشد و با سرعت آهسته تر حرکت

کند و بر عکس وقتی تعداد موانع اطراف ربات کم باشد لازم است برد توجه ربات و همچنین سرعت حرکت آن افزایش یابند. باین ترتیب وقتی تراکم موانع زیاد است، ربات به آنها نزدیک شده و از میانشان عبور می نماید. در چنین حالتی چون خطر برخورد زیادتر است حرکت ربات آهسته تر صورت می گیرد. بر عکس وقتی تراکم موانع در اطراف ربات کم است، ربات بیشتر به جستجوی نواحی بدون مانع می پردازد و با توجه به کمتر بودن خطر برخورد سریع تر حرکت می نماید.

فرض کنید A_mg برد توجه و حداکثر سرعت مجاز ربات باشند برای تنظیم این دو مقدار از اتوماتون یادگیر استفاده می نماییم.

ورودی اتوماتون β یا مقدار تنبیه می باشد. این مقدار با کمک تابع F که آرگومان های آن از سنسورها گرفته می شود، مشخص می گردد. هر آرگومان یک مقدار با پیری است. این مقدار با پیری مشخص می کند که آیا سنسور مربوطه در حوزه دید خود مانع را تشخیص داده است یا نه. اتوماتون مورد استفاده دارای دو عمل مختلف می باشد. یک عمل Increase جهت افزایش مقادیر A_mg و MP_Speed و دیگری Decrease جهت کاهش این مقادیر می باشد. هر عمل یک مقدار احتمال مربوط به خود دارد که با p_i نشان داده می شود. این مقدار احتمال در هر قدم زمانی با توجه به عمل انتخاب شده و تنبیه دریافت شده به کمک رابطه زیر بروز در آورده میشود:

$$IF \alpha(n) = \alpha_i, \quad (2)$$

$$WHEN \beta=0$$

$$p_i(n+1) = (1-a).p_i(n), \text{ for all } j \neq i$$

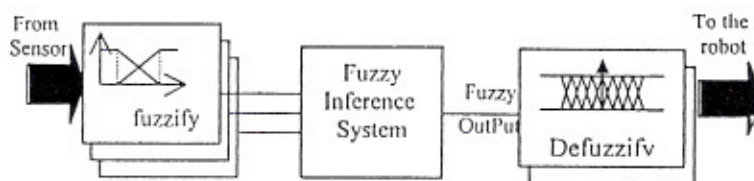
$$p_i(n+1) = p_i(n) + a.[1 - p_i(n)]$$

$$WHEN \beta=1$$

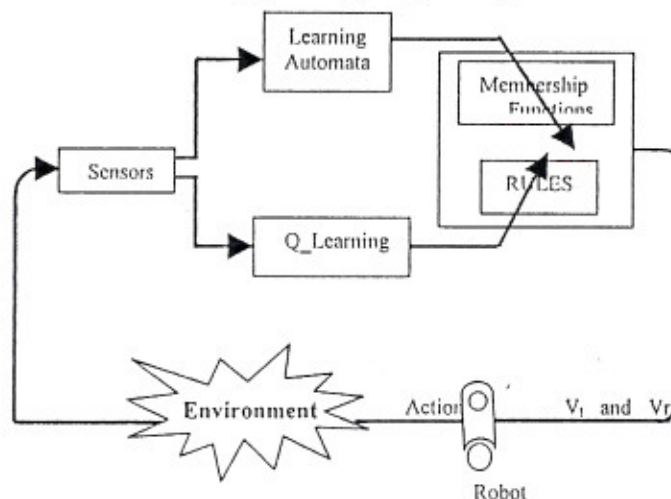
$$p_i(n+1) = b/(r-1) + (1-b).p_i(n), \text{ for all } j \neq i$$

$$p_i(n+1) = (1-b).p_i(n)$$

احتمال هر دو عمل در آغاز کار با هم مساوی و هر کدام برابر 0.5 انتخاب شده اند. پارامترهای a و b بطور تجربی مساوی با 0.4 در نظر گرفته شده اند. قدم زمانی r و تعداد عمل های اتوماتون می باشند.



شکل ۵ معماری کنترل کننده فازی



شکل ۶ نحوه ارتباط کنترل کننده فازی، اتوماتون تصادفی یادگیر و یادگیری Q

عملی که در یک شرایط خاص منجر به شکست شود، احتمال آن با کمک رابطه (۲) کاهش می یابد و شانس کمتری برای انتخاب خواهد داشت و برعکس. به این ترتیب با توجه به تراکم موانع در اطراف ربات در هر قدم زمانی می توان Δ_rng و MP_Speed مناسب را یافت. از Δ_rng جهت مشخص کردن پارامترهای توابع عضویت فازی ورودی و از MP_Speed جهت مشخص کردن پارامترهای توابع عضویت فازی خروجی استفاده می شود (شکل های ۳ و ۴).

۵) شرح الگوریتم یادگیر

هدف الگوریتم یادگیر یافتن نگاشتی مناسب از مجموعه حالت های محیط به مجموعه عملیات مجاز و ممکن برای ربات می باشد. گفتیم که در شبیه سازی از ربات کیپرا استفاده نموده ایم، این ربات ۸ سنسور دارد که در آزمایشات تنها سه سنسور آن، جهت تشخیص موانع در جلو، سمت راست و سمت چپ ربات به کار می روند. هر سنسور فاصله تا نزدیک ترین مانع به ربات را در یک جهت خاص اندازه گیری می نماید. مقدار اندازه گیری شده با توجه به توابع عضویت فازی

مقدار تنبیه β با کمک رابطه زیر به دست می آید:

$$IF \alpha(n) = 'Increase' \quad (3)$$

$$IF(ActiveSensorNum \geq 1/3.TotalSensorNum)$$

$$\beta = 1$$

$$ELSE IF \alpha(n) = 'Decrease'$$

$$IF(ActiveSensorNum < 1/3.TotalSensorNum)$$

$$\beta = 1$$

$$ELSE$$

$$\beta = 0$$

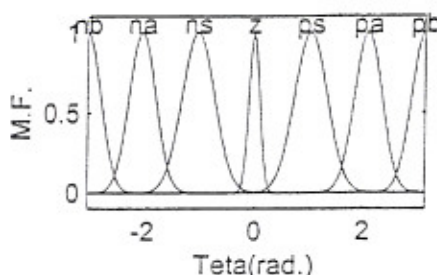
ActiveSensorNumber تعداد سنسورهایی است

که حداقل یک مانع در حوزه دید خود تشخیص داده اند و Total Sensor Number تعداد کل سنسورهای ربات می باشد. در هر قدم زمانی اتوماتون با توجه به احتمال هر عمل و باروش رولوت ویل، عملی را جهت اجرا انتخاب می نماید. این عمل بر روی برد توجه و حداکثر سرعت مجاز تاثیر گذاشته، مقدار آنها را به ترتیب به اندازه $0.4/0.5$ و $0.5/0.7$ میلی متر در دنیای واقعی، تغییر می دهد.

می شود، بدون اینکه در کارایی الگوریتم اشراقابل ملاحظه ای داشته باشد.

۶ رفتار هدف یابی

ناوبری در حقیقت ترکیب دو رفتار پرهیز از مانع و هدف یابی است. برای هدف یابی نیز از روشی مشابه پرهیز از مانع یعنی سیستم فازی و یادگیری Q استفاده می نماییم. برای مشخص کردن حالت های ورودی از زاویه میان ربات و هدف استفاده می شود. این زاویه در فاصله $[-\pi, \pi]$ محاسبه می گردد. چرخش خلاف جهت عقربه های ساعت با مقدار مثبت و چرخش در جهت حرکت عقربه های ساعت با مقدار منفی نشان داده می شود (جهت مثلثاتی استاندارد). زاویه میان ربات و هدف می تواند هر یک از مقادیر فازی شکل ۷ را اختیار نماید.



شکل ۷) توابع عضویت فازی ورودی برای رفتار هدف یابی نگاشت میان مجموعه حالات و مجموعه اعمال مجاز توسط یادگیری Q انجام می گیرد. ابتدا مقدار Q برای تمام زوج های حالت و عمل مجاز یک در نظر گرفته می شود. یک الگوریتم انتخاب، در ۲۰ درصد از مواقع بطور تصادفی (exploration) و در ۸۰ درصد یک از حالات ممکن انتخاب می کند. (مرحله انتخاب قوانین فازی) ربات این عمل را در محیط اجرا می کند و بازای آن پاداش یا تنبیهی دریافت می دارد. که برای بروز در آوردن مقادیر $Q(s,a)$ با کمک رابطه (۵) استفاده می شود. γ, α هر دو مساوی 0.9 انتخاب شده اند. در رفتار هدف یابی میزان پاداش متناسب با کاهش زاویه میان جهت حرکت ربات و هدف تعریف میشود. در صورتی که این زاویه کوچکتر از $\pi/8$ باشد پاداش زیادتری دریافت می گردد.

ورودی برای هر سنسور می تواند در یکی از دو ناحیه dangerous یا safe واقع شود. بنابر این ربات نسبت به موانع در ۸ وضعیت مختلف ممکن است قرار گیرد. حال باید مشخص کرد که در هر وضعیت چه عملی مناسب تر است.

الگوریتم یادگیری استفاده شده Q-Learning می باشد. برای حالتی که هر سه طرف ربات در ناحیه خطرناک تشخیص داده شده اند، عمل مناسب چرخش در جا در نظر گرفته شده است. برای سایر حالات الگوریتم یادگیر عمل مناسب را تعیین می نماید.

در ابتدا مقدار Q برای تمام زوج های حالت و عمل مجاز یک در نظر گرفته می شود. یک الگوریتم انتخاب، در ۲۰ درصد از مواقع بطور تصادفی (exploration) و در ۸۰ درصد از مواقع با توجه به مقدار Q، عملی را جهت انجام در هر یک از حالات ممکن انتخاب می نماید. (مرحله انتخاب قوانین فازی) ربات این عمل را در محیط اجرا می کند و بازای آن پاداش یا تنبیهی دریافت می دارد، که برای بروز در آوردن مقادیر $Q(s,a)$ استفاده می گردد.

















برای رفتار پرهیز از مانع مقدار پاداش بر اساس میزان افزایش نزدیک ترین فاصله اندازه گیری شده تا مانع توسط هر حسگر ربات، تعیین میگردد. حسگر جلواهمیت بیشتری دارد، بنابراین به آن وزن بزرگتری داده می شود، و اگر در جهت این حسگر فاصله تا مانع افزایش یابد، پاداش بیشتری دریافت خواهد شد.

$$\begin{aligned} \Delta d_i &= new_d_i - prev_d_i, i = 1, 2, 3 \\ r &= \sum_{i=1}^3 w_i \cdot \Delta d_i \\ w &= \{1, 10, 1\} \end{aligned} \quad (4)$$

Δd_i میزان افزایش نزدیک ترین فاصله اندازه گیری شده تا مانع توسط حسگر i ام ربات می باشد و وزن w_i اهمیت آن حسگر است. برای بروز در آوردن مقادیر $Q(s,a)$ از رابطه زیر استفاده می گردد. γ و α هر دو برابر 0.9 انتخاب شده اند.

$$\begin{aligned} Q(s,a) &= Q(s,a) + \alpha[r + \gamma V(y) - Q(s,a)] \quad \alpha \in (0,1] \\ V(y) &= \max_a Q(x,a) \end{aligned}$$

الگوریتم شرح داده شده با دقت یک رقم اعشار پس از ۴۸۰ قدم زمانی با تعداد ۴۰ برخورد همگرا می شود اما اگر از قسمت اعشار صرف نظر کنیم با تنها ۲۷۰ قدم زمانی و ۲۰ برخورد الگوریتم همگرا

1)  	teta=-pi , teta=pi	5)  	teta=0
2)  	-pi<teta<-pi/2	6)  	0<teta<pi/2
3)  	teta=-pi/2	7)  	teta=pi/2
4)  	-pi/2<teta<0	8)  	pi/2<teta<pi

شکل ۸ نمونه هایی از حالات دو ربات همکار نسبت به هم

۷) همکاری

در اینجا نیز از روشی مشابه قبل یعنی سیستم فازی و یادگیری Q استفاده می نماییم با این تفاوت که نگاشتمیان حالات محیط و اعمال مجاز توسط الگوریتم یادگیری Q توزیع شده انجام می گیرد.

برای مشخص کردن حالت های ورودی از دو معیار استفاده می نماییم . یکی همسایگی است که مشخص می کند آیا سنسور جلو ربات، وجود ربات دیگر را حس کرده است یا نه. اگر حس نکرده باشد ربات می تواند از همان الگوریتم ناوبری ساده استفاده نماید و نیازی به همکاری نیست . اما اگر این سنسور وجود ربات دیگر را تشخیص داده باشد ، وضعیت حساس تری دارد و باید به گونه ای متفاوت عمل کند. چرا که ربات دوم خود قادر به تصمیم گیری ، حرکت و تغییر شرایط محیط است. از طرفی این ربات دوم هم در محیط وظایفی دارد که اگر نتواند به آنها عمل کند عملکرد کل سیستم کاهش می یابد. معیار دوم زاویه میان راستای حرکت ربات و رباتی که در جلوی خود تشخیص داده است، می باشد.

الگوریتم یادگیری Q توزیع شده برای بروز در آوردن مقادیر $Q(s,a)$ از رابطه زیر استفاده می نماید:

$$Q(s,a) = (1-\alpha)Q(s,a) + \alpha(R(s,a) + \gamma \sum_j f(i,j)V_j(s'_j))$$

$$V_j(s'_j) = \max_a Q(s'_j,a)$$

وقتی تعداد نودها (ربات ها) دو باشد و وزن های $f(1,1)=f(1,2)=f(2,2)=1$ (مثلاً برای ربات ۱) به شکل زیر تبدیل می گردد. α, γ برابر ۰/۹ . انتخاب میشوند.

$$Q(s,a) = (1-\alpha)Q(s,a) + \alpha(R(s,a) + \gamma(V_1(s'_1) + V_2(s'_2)))$$

$$V_j(s'_j) = \max_a Q(s'_j,a)$$

يك الگوریتم انتخاب در ۲۰ درصد از مواقع بطور تصادفی (exploration) و در ۸۰ درصد از موارد با توجه به مقادیر Q عملی را جهت انجام در هر يك از حالات ممکن انتخاب می کند. (مرحله انتخاب قوانین فازی) ربات این عمل را در محیط اجرا می کند و بازای آن پاداش یا تنبیهی دریافت می دارد، که برای بروز در آوردن مقادیر Q استفاده می شوند.

در اینجا چون از الگوریتم یادگیری توزیع شده استفاده می شود علاوه بر پاداش و حالت بعدی خود ربات، پاداش و حالت بعدی ربات دیگر نیز در بروز در آوردن مقادیر Q مورد استفاده قرار می گیرند.

برای مشخص کردن مقدار تنبیه یا پاداش از دو معیار استفاده می شود. یکی اینکه فاصله ربات تا ربات دیگری که در جلو خود حس کرده کاهش یا افزایش یابد، که متناسب با میزان این کاهش یا افزایش، تنبیه یا پاداش دریافت می دارد. دیگری اینکه دو ربات با یکدیگر برخورد کنند که در این صورت يك تنبیه ثابت مساوی ۱۰- در نظر گرفته می شود.

۸) ترکیب رفتارهای همکاری ، پرهیز از مانع و هدف یابی

باز هم بحث ترکیب کردن رفتارهای پرهیز از مانع و هدف یابی مطرح است. اما این بار وضعیت دو ربات نسبت به هم نیز در تصمیم گیری برای انتخاب عمل مناسب بسیار مهم است. برای حل مساله از راه حلی شبیه به ناوبری استفاده می نماییم. به این ترتیب که همکاری و پرهیز از مانع از اولویت بالاتری نسبت به هدف یابی برخوردارند. حال اگر در يك قدم زمانی خاص بتوان عملی یافت که برای هر سه منظور مناسب باشد، آن عمل را انتخاب کرده انجام می دهیم. در غیر

ترکیب رفتارهای مختلف موضوعی مفصل و گسترده تحت عنوان Behavior Blending و Behavior Fusion می باشد. ما برای ترکیب رفتارها از روشی بسیار ساده و بسیار کار آمد استفاده نموده ایم. به این ترتیب که به طور ثابت رفتار پرهیز از مانع از اولویت بالاتری نسبت هدف یابی برخوردار است. حال اگر در یک قدم زمانی خاص می توان عملی مناسب برای هر دو رفتار انتخاب نمود، آن عمل را انتخاب کرده و انجام می دهیم. در غیر این صورت عملی را انجام می دهیم که برای رفتار پرهیز از مانع مناسب باشد. این کار با کمک یک ماکزیم گیری ساده انجام می شود.

$$Q = \min(\text{avoid}Q(s_j, a_i), \text{target}Q(t_n, a_i))$$

$$\text{nav}Q(s_j, a_i) = \begin{cases} Q & \text{if } Q \geq \tau \\ \text{avoid}Q(s_j, a_i) & \text{otherwise} \end{cases} \quad (9)$$

avoidQ مقادیر Q به دست آمده برای رفتار پرهیز از مانع، targetQ مقادیر Q به دست آمده برای رفتار هدف یابی و navQ مقادیر حاصل برای رفتار ناوبری می باشند. با مشخص شدن مقادیر Q برای رفتار ناوبری، قوانین فازی مناسب در هر قدم زمانی تعیین شده، رفتار ناوبری با موفقیت انجام می پذیرد.

جدول ۱ و شکل ۹ نشانگر اهمیت به کارگیری اتوماتون یادگیر برای تطبیقی کردن کنترل کننده می باشند. در این جدول D_{\max} خطر ماکزیم، D_{average} میانگین خطری و Moving Length طول مسیر می باشد. خطر ماکزیم طبق رابطه زیر محاسبه می گردد:

$$D = \sum_{j=1}^n (1.0 - \frac{x_j(t)}{A - \text{rng}(t)}) \quad (10)$$

$$D_{\max} = \text{Max}_i(D)$$

که در آن D مقدار خطر در لحظه t، $x_j(t)$ فاصله اندازه گیری شده تا نزدیک ترین مانع به ربات، توسط سنسور زام و n تعداد سنسورهای ربات می باشند. t قدم زمانی و D_{\max} ماکزیم خطر است. به عبارتی ماکزیم خطر، بیشترین D اندازه گیری شده طی مسیر، در قدم های زمانی مختلف می باشد.

این صورت عملی را انجام می دهیم که برای دو منظور همکاری و پرهیز از مانع مناسب باشد.

این کار با کمک تابع nav2rob-choose انجام شده است. در هر قدم زمانی ابتدا وضعیت ربات نسبت به هدف (gstate) و نیز وضعیت ربات نسبت به دیگر (cstate) مشخص می گردند. gstate، cstate، حالتی هستند که بیشترین شباهت را به حالت جاری دارند. (حالت جاری در این حالت ها بزرگترین درجه عضویت را دارد.)

$$Q1() = \min(Q(gstate, a_n), Q(ObstacleAvoidState_n, a_n))$$

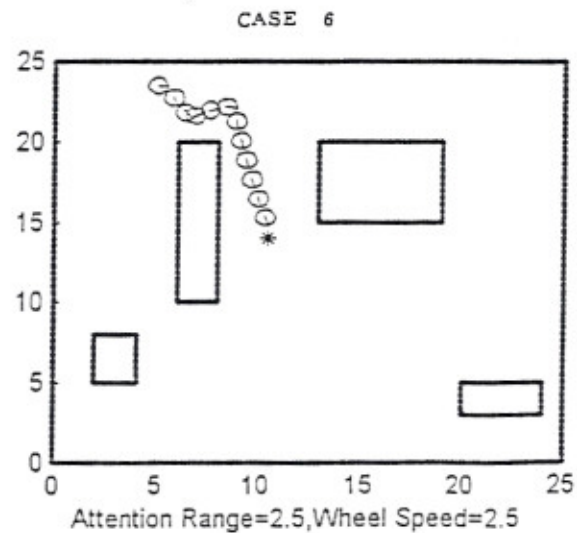
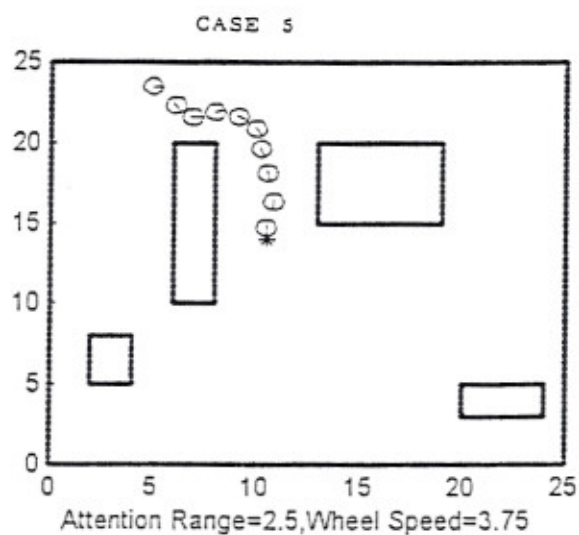
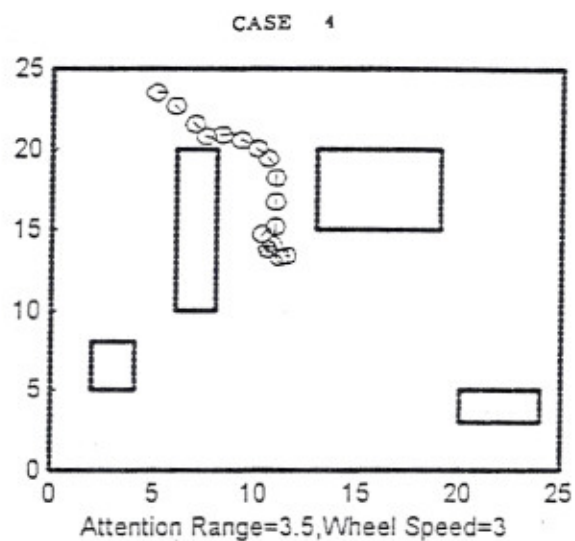
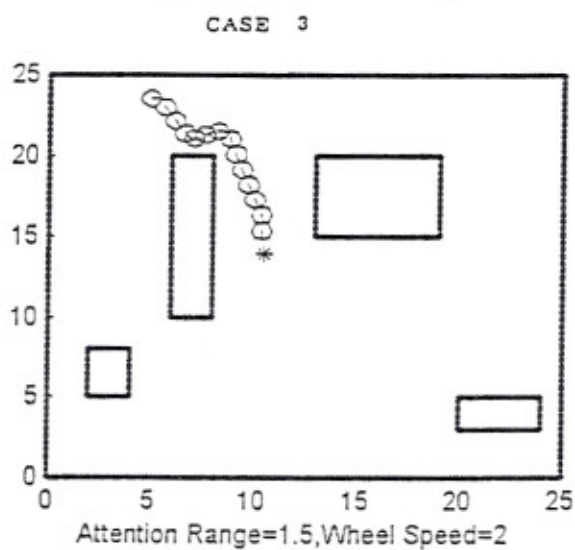
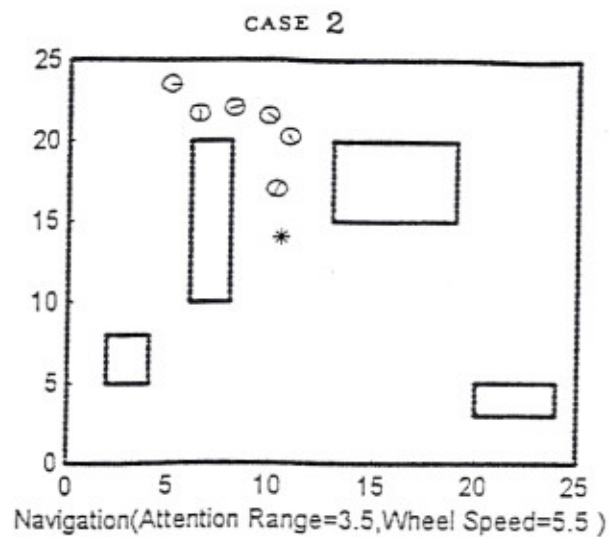
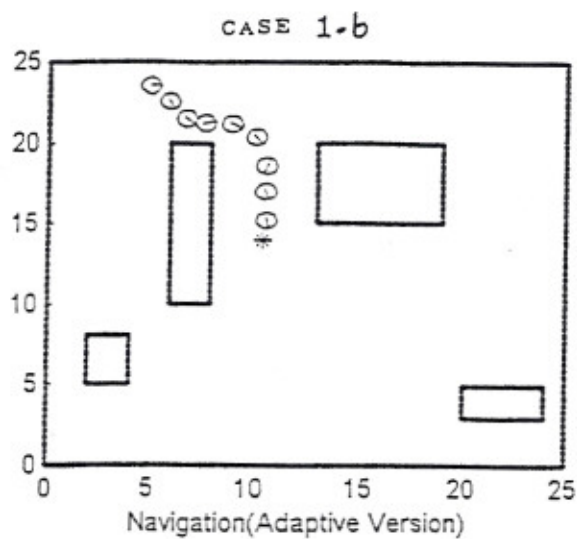
$$Q2() = \min(Q(gstate, a_n), Q(ObstacleAvoidState_m, a_n))$$

$$Q3() = \min(Temp1, Temp2) \quad \text{for } n=1, \dots, 25, m=1, \dots, 8$$

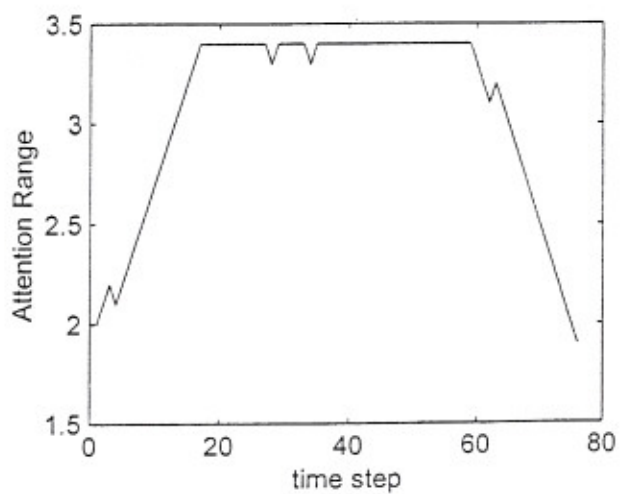
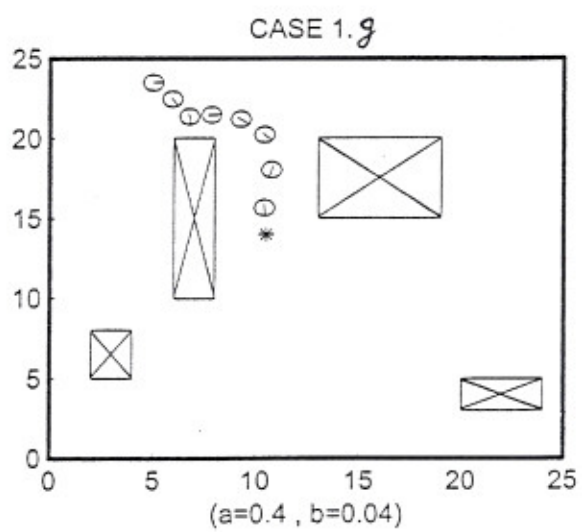
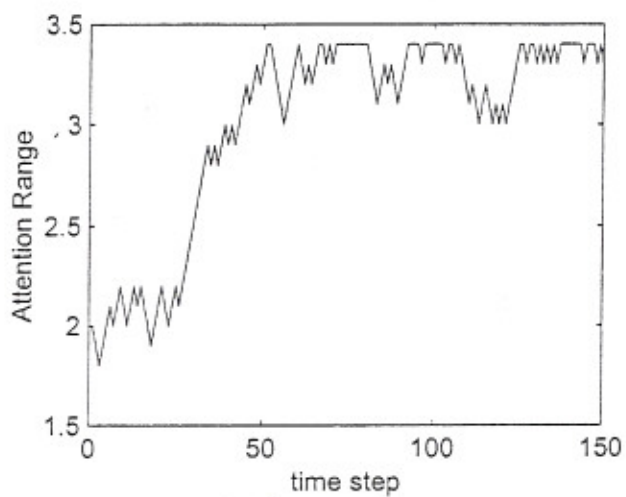
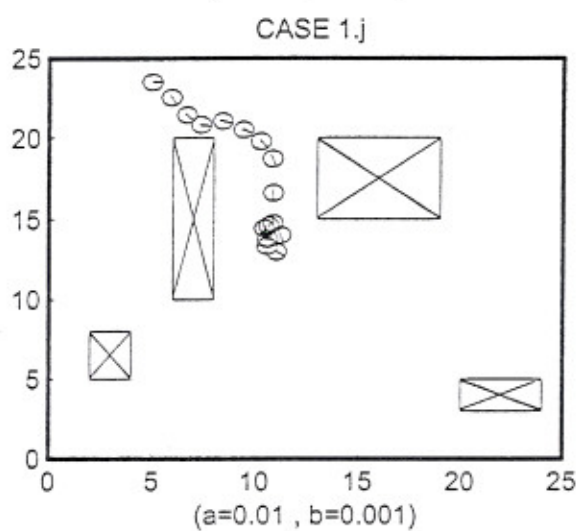
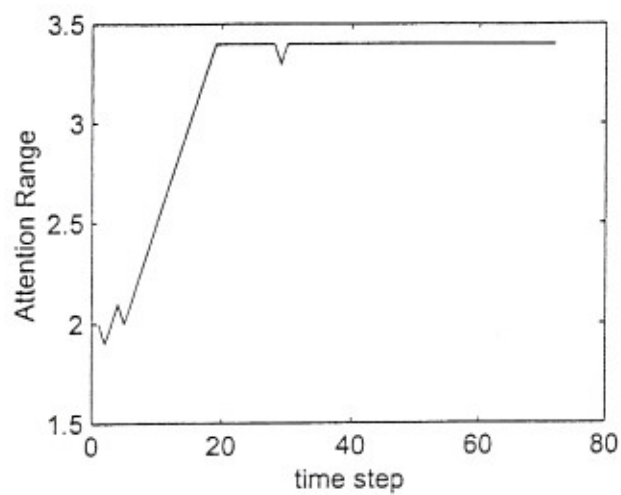
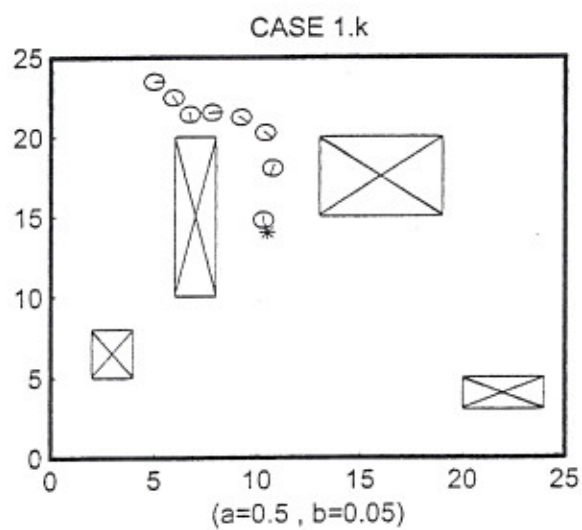
در اینجا نیز یک الگوریتم انتخاب در ۲۰ درصد از مواقع به طور تصادفی (exploration) و در ۸۰ درصد از موارد با توجه به ماکزیم $Q2()$ یا $Q3()$ قوانین فازی را انتخاب می نماید. باین ترتیب که اگر ماکزیم $Q3()$ از یک مقدار آستانه بیشتر باشد، عمل مناسب برای هر سه رفتار را ارائه می دهد. در غیر این صورت ماکزیم $Q2()$ بررسی می شود و جهت انتخاب عمل مناسب استفاده می گردد.

۹ بررسی عملکرد کنترل کننده طراحی شده

ناوبری عبارت است از حرکت ربات در یک محیط به سمت هدف در حالیکه از موانع پرهیز می کند. برای حل مساله ناوبری دو روش کلی موجود است. روش اول اینکه هر دو رفتار پرهیز از مانع و هدف یابی توأماً و همزمان صورت می گیرد. ایراد این روش بزرگتر شدن فضای جستجو و کند تر شدن همگرایی می باشد. روش دوم (که ما از این روش استفاده می کنیم) این است که هر یک از دو رفتار پرهیز از مانع و هدف یابی بطور جداگانه بررسی و یاد گرفته شود و سپس با ترکیب این دو رفتار، ناوبری تحقق یابد. مثلاً اگر هدف در سمت راست ربات باشد و مانع در جلو آن، رفتار پرهیز از مانع ربات را از جلو رفتن منع می کند و رفتار هدف یابی آن را به سمت راست هدایت می کند، به این ترتیب ربات نهایتاً به راست می رود و ناوبری انجام می پذیرد. اما ترکیب این دو رفتار همیشه به این سادگی نیست مثلاً حالتی را در نظر آورید که مانع و هدف هر دو در مقابل ربات قرار دارند و مانع میان ربات و هدف می باشد.



شکل ۹) اهمیت اتوماتون تصادفی یادگیر در نحوه عملکرد کنترل کننده



CASE	A-rng	Speed	S.L.A.param a	S.L.A.param b	D _{average}	MovingLength	D _m	TimeStep
1.a	Adaptive	Adaptive	0.2	0.2	0.1088	13.0527	0.8234	83
1.b	Adaptive	Adaptive	0.4	0.4	0.1095	12.6224	0.8678	78
1.r	Adaptive	Adaptive	0.5	0.5	0.1179	13.5473	0.8235	84
1.c	Adaptive	Adaptive	0.6	0.6	0.1211	13.0256	0.8235	83
1.d	Adaptive	Adaptive	0.8	0.8	0.1006	17.6195	0.8235	108
1.e	Adaptive	Adaptive	0.1	0.01	0.1343	13.1003	0.8705	78
1.f	Adaptive	Adaptive	0.2	0.02	0.1334	20.8869	0.8398	116
1.g	Adaptive	Adaptive	0.4	0.04	0.1250	13.1157	0.8235	76
1.k	Adaptive	Adaptive	0.5	0.05	0.1434	13.0741	0.8235	72
1.h	Adaptive	Adaptive	0.6	0.06	0.1354	12.6825	0.8235	71
1.i	Adaptive	Adaptive	0.8	0.08	0.1021	14.6554	0.8235	95
1.j	Adaptive	Adaptive	0.01	0.001	0.1254	21.5008	0.8833	*175
1.l	Adaptive	Adaptive	0.02	0.002	0.1208	13.0323	0.8235	103
1.m	Adaptive	Adaptive	0.04	0.004	0.1414	17.5586	0.8235	106
1.r	Adaptive	Adaptive	0.05	0.005	0.1449	12.8454	0.8423	78
1.n	Adaptive	Adaptive	0.06	0.006	0.1409	12.9513	0.8235	77
1.o	Adaptive	Adaptive	0.08	0.008	0.1427	13.1618	0.8435	80
2	3.5	5.5	None	None	0.1755	13.8931	0.8571	59
3	1.5	2	None	None	0.0539	12.1369	0.6340	139
4	3.5	3	None	None	0.1466	16.4693	0.9445	*178
5	2.5	3.75	None	None	0.0978	12.9748	0.6	92
6	2.5	2.5	None	None	0.1060	12.6233	0.7	118

*: can't find goal in 150 time steps.

Table 1.The obtained results of the various cases examined in fig. 9

این میانگین با کمک رابطه زیر محاسبه می شود:

$$D_{average} = \frac{1}{P_{time}} \cdot \frac{1}{n} \cdot \sum_{i=1}^{P_{time}} \sum_{j=1}^n \left(1.0 - \frac{x_i(t)}{A - rng(t)} \right) \quad (11)$$

P_{time} کل قدم های زمانی لازم، از لحظه شروع حرکت تا لحظه رسیدن به هدف می باشد و $A - rng(t)$ برد توجه در قدم زمانی t است. یادقت در شکل ۹ و توجه به جدول ۱ می توان به نتایج زیر دست یافت:

۱- در صورت عدم استفاده از اتوماتون یادگیر (کنترل کننده غیر تطبیقی) یا انتخاب نامناسب پارامترهای آن، مسیر ناممکن خواهد شد (CASE ۱_b و ۲_b)، یا میزان خطر طی مسیر افزایش خواهد یافت، (CASE ۱_k یا ۴ یا ۳_b و CASE ۱_b)، یا رسیدن به هدف با تاخیر انجام خواهد شد و ربات دیرتر به هدف خود خواهد رسید، (۱_j یا ۳ یا ۶_b و CASE ۱_b).
۲- اگر چه می توان بازای بعضی مقادیر خاص برای A_{mg} یا MP_Speed به نتایج خوبی دست یافت (CASE ۵)، اما اولاً این نتایج به خوبی حالت تطبیقی نخواهند بود و ثانیاً یافتن آنها عملی پر هزینه و زمانبر خواهد بود. باین ترتیب اهمیت اتوماتون یادگیر که می تواند در هر قدم زمانی مقدار مناسب این متغیرها را بطور اتوماتیک محاسبه کند آشکار تر می گردد.

شکل (۱۰) نمونه ای از رفتار ناوبری یک ربات متحرک در یک محیط خاص را نشان می دهد. ربات پس از ۳۳۹ قدم زمانی (هر قدم زمانی ۱/۰ ثانیه می باشد) مسیری به طول ۳۳/۳ را طی می نماید. ماکزیمم خطر ۱/۲۸ و خطر میانگین ۰/۱۳ اندازه گیری شده اند. ربات برای رسیدن به هدف خود مسیری مناسب، کوتاه و هموار انتخاب می نماید و به خوبی به هدف خود می رسد. شکل ۱۱ مثال دیگری را نشان می دهد. به نمودار هر شکل توجه کنید، این نمودار تغییرات برد توجه ربات را در حین حرکت نشان می دهد. این تغییرات حاصل عملکرد اتوماتون یادگیر می باشد.

شکل های (۱۲) و (۱۳) نحوه عملکرد دو ربات در یک محیط را نشان می دهند. در این شکل ها r نماینده ربات (robot)، g نماینده هدف (goal) و نقطه شروع (start) می باشند. هدف یک مربوط به ربات یک می باشد و با علامت 'نشان داده شده است.

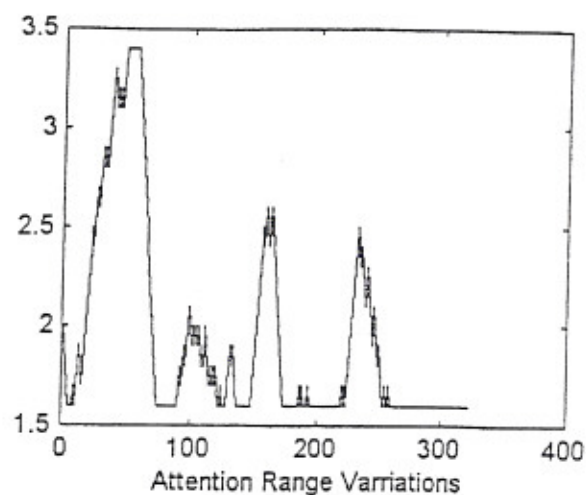
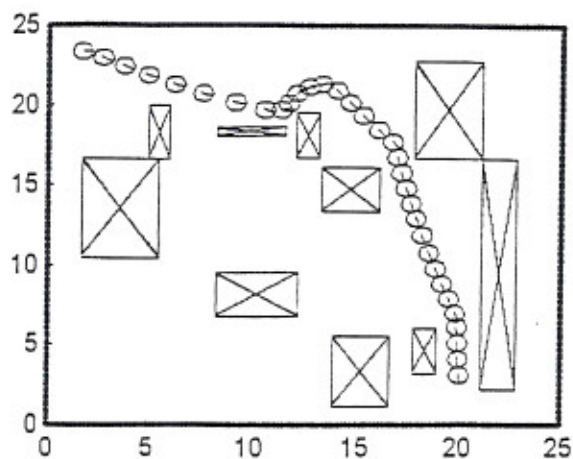
هدف دو نیز مربوط به ربات دو بوده و با علامت '*' نشان داده شده است. ربات ها را پس از هر ۱۰ قدم زمانی در محیط نمایش داده ایم.

در شکل ۱۲ دو ربات، در حالت ۵، از مجموعه حالت های تعریف شده در شکل (۸) قرار داده شده اند. یعنی ϕ_1 و ϕ_2 (زاویه میان راستای هر ربات با افق در جهت مثبت استاندارد) هر دو صفر هستند. همان طور که دیده می شود، در این حالت ربات ۲، راه ربات ۱ را سد نموده است. ولی ربات ۱، تزاحمی برای ربات ۲ ندارد. ربات ۲، ابتدا بسیار آهسته حرکت می نماید و در عین حال سعی دارد ربات ۲ را از پایین دور بزند چرا که در بالا با مانع مواجه است (شکل ۱۲- a). رفته رفته ربات ۲ به سمت هدف خود می رود و از مسیر ۲۱ خارج می شود. با کنار رفتن ربات ۲ از سر راه ۲۱ هر ربات مسیر خود را به سمت هدفش طی می کند. شکل ۱۲- f تغییرات برد توجه را برای هر یک از ربات ها نمایش می دهد. وقتی که شرایط موقعیت ربات نسبت به موانع ثابت باشد، برد توجه هم ثابت می شود. برعکس، با تغییرات مداوم محیط و میزان پراکندگی موانع، تغییرات برد توجه هم سریع تر می شود.

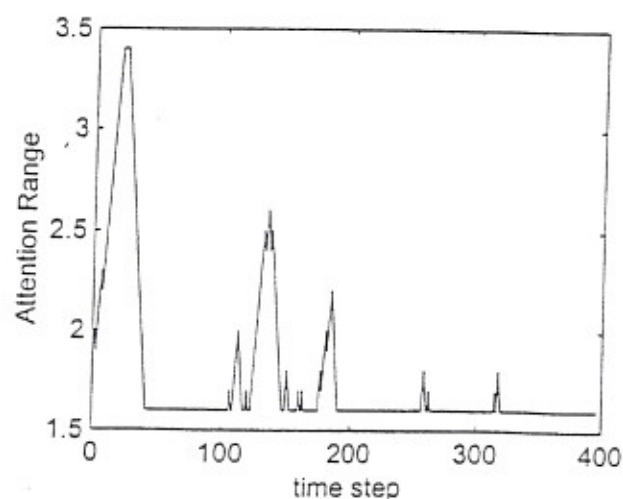
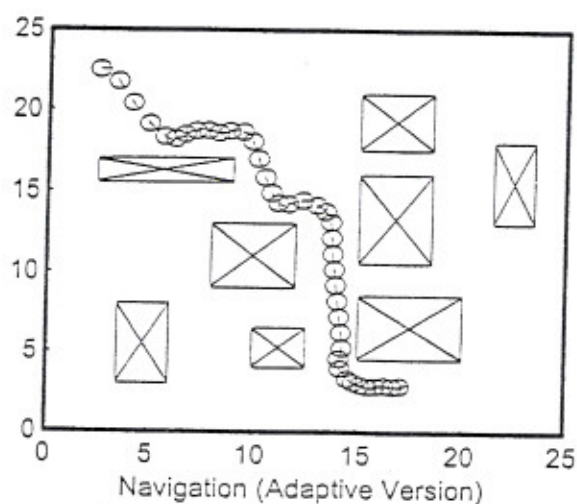
در شکل ۱۳ دو ربات در حالت ۸، از مجموعه حالت های تعریف شده در شکل ۸ قرار داده شده اند. یعنی ϕ_1 برابر صفر و ϕ_2 برابر $\pi/6$ می باشند. در این حالت هر دو ربات راه یکدیگر را سد نموده اند.

بنابر این چون هر ربات برای عبور به سمت هدف خود با ربات دیگر مواجه می شود حرکت ربات ها به آهستگی انجام می شود. (شکل ۱۳- a). پس از عبور ۲۱ مسیر حرکت برای ۲۱ باز و برای ۲۲ نسبتاً باز می شود. پس ۲۱ با سرعت بیشتر به سمت هدف خود پیش می رود، ولی ۲۲ که هنوز مسیر خود را کاملاً باز نمی باید آهسته تر حرکت می نماید (شکل ۱۳- b). در آخر با خارج شدن ۲۱ از مسیر ۲۲، هر دو ربات به سمت هدف خود پیش می روند (شکل ۱۳- c).

ربات ۲۱ پس از ۵۹۰ قدم زمانی و ربات ۲۲ پس از ۳۹۰ قدم زمانی به هدف خود می رسند. هر قدم زمانی ۰/۱ ثانیه می باشد. طی این مسیر خطر ماکزیمم برای ۲۱ مقدار ۱/۵ و برای ۲۲ برابر ۰/۸۴ می باشد. میانگین خطر نیز برای ۲۱ مساوی ۰/۱۱ و برای ۲۲ برابر ۰/۰۷ می باشد. شکل (۱۳- d) تغییرات برد توجه را برای ربات ها نمایش می دهد.



شکل ۱۰) ناوبری یک ربات متحرک در یک محیط نمونه با استفاده از کنترل کننده فازی تطبیقی

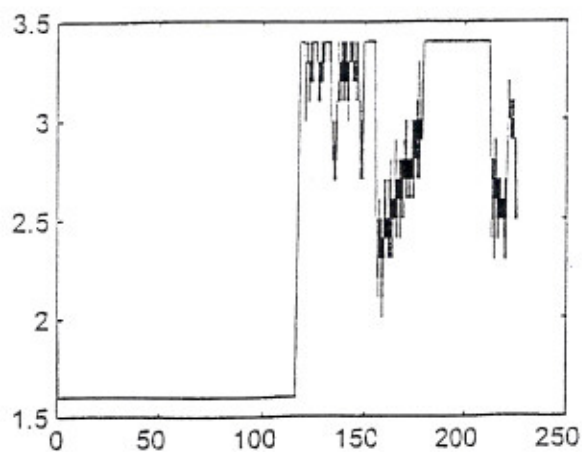
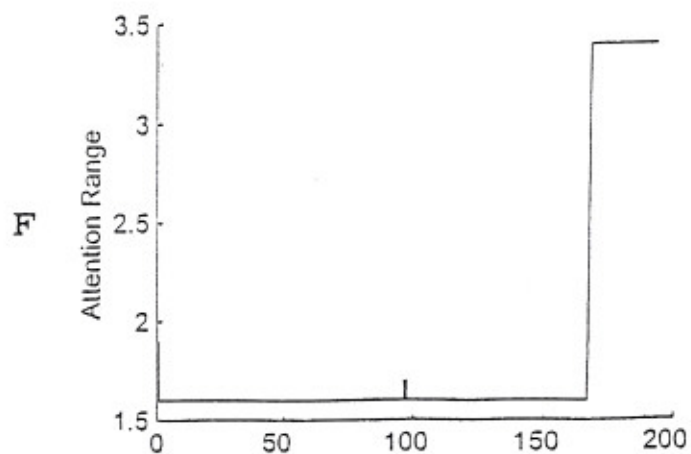
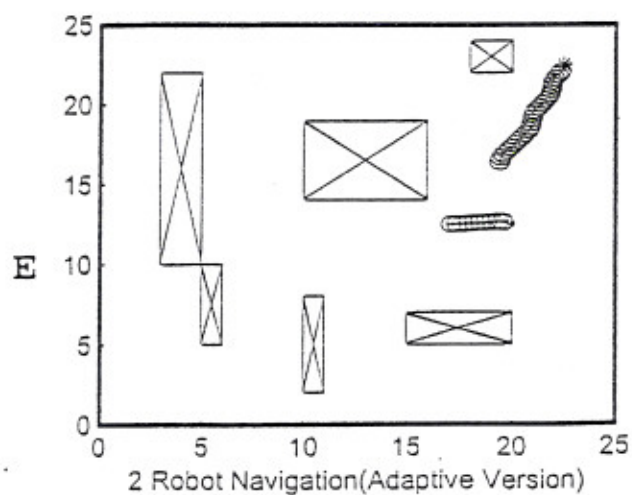
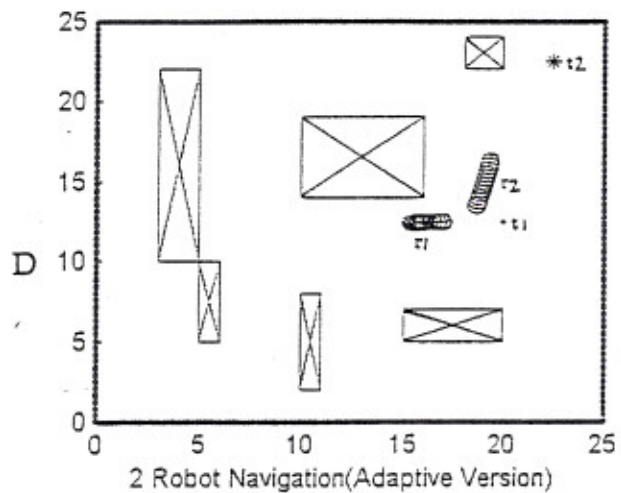
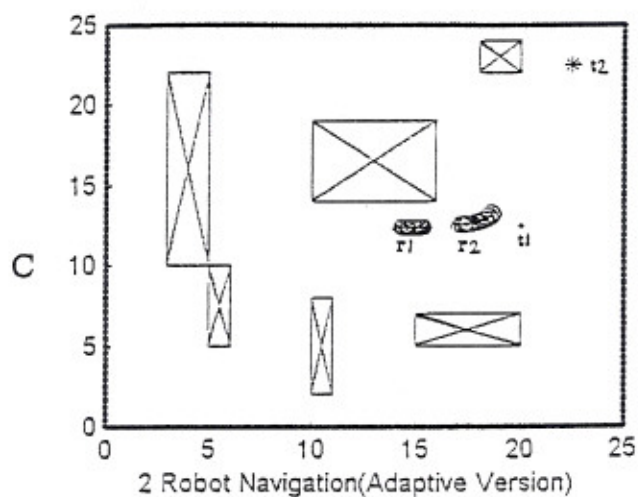
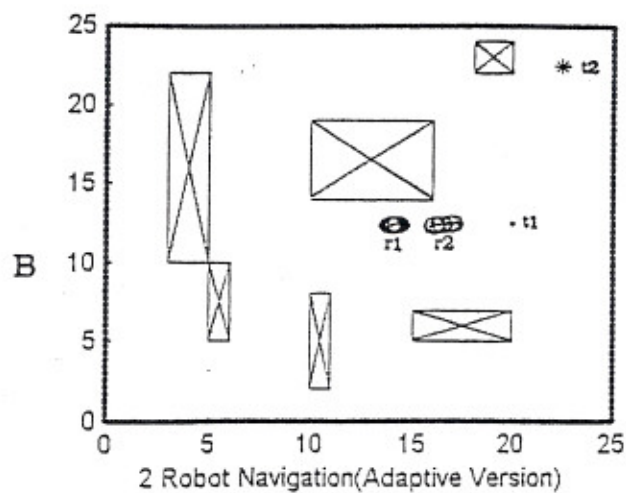
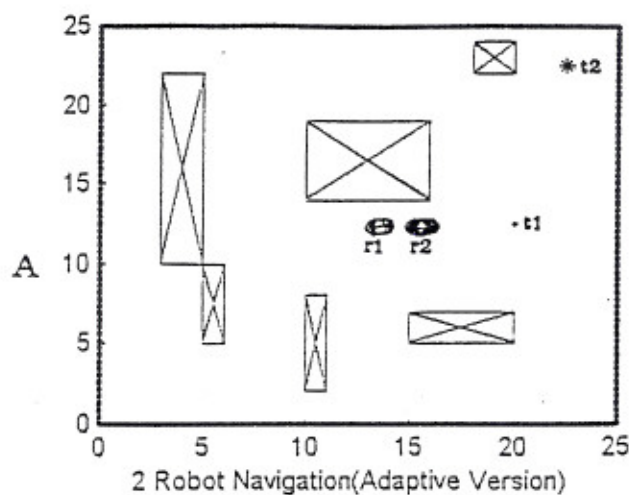


شکل ۱۱) ناوبری یک ربات متحرک در یک محیط نمونه با استفاده از کنترل کننده فازی تطبیقی

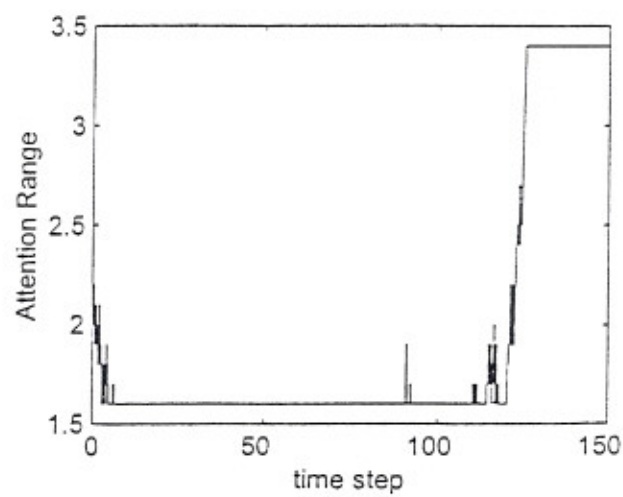
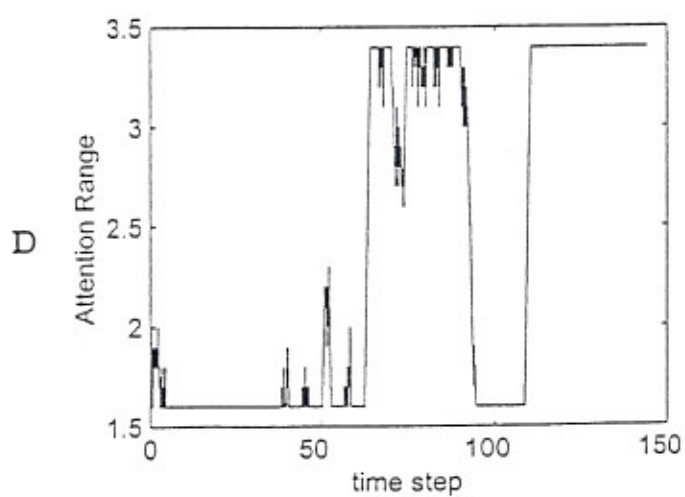
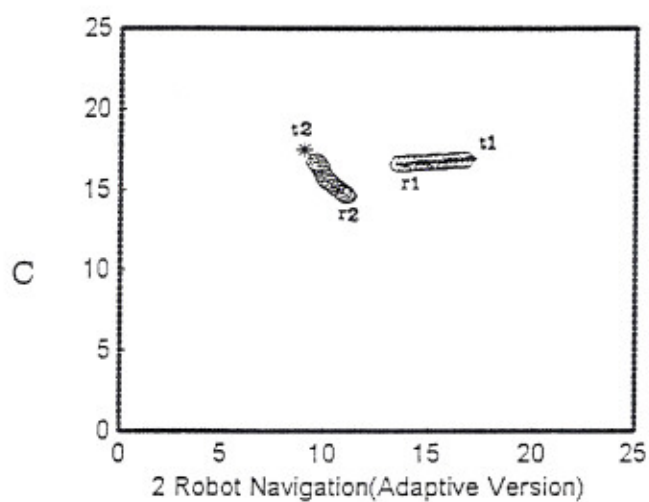
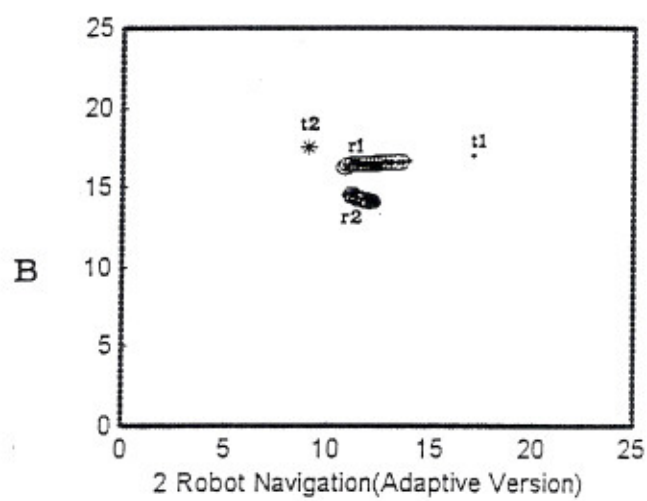
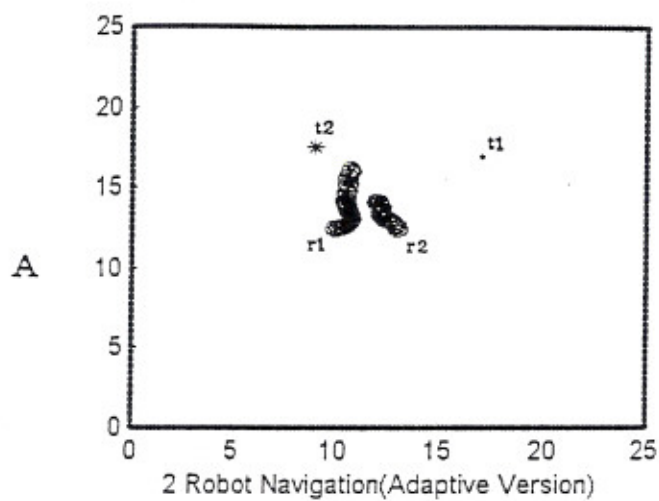
Figure	A-rng	Speed	$D_{average}$	MovingLength	D_m	TimeStep
10	Adaptive	Adaptive	0.13	33.3	0.98	339
11	Adaptive	Adaptive	0.12	32.02	0.79	443

Table) Obtained Results

جدول ۲) نتایج اندازه گیری شده



شکل ۱۲: ناوبری دو ربات متحرک در یک محیط نمونه با استفاده از کنترل کننده فازی تطبیقی



شکل ۱۳) ناوبری دوربات متحرک در یک محیط نمونه با استفاده از کنترل کننده فازی تطبیقی

۹) نتایج و پیشنهادات

در يك محيط ديناميك كه دائماً در حال تغيير است، استفاده از روش های سراسری غيرممکن يا بسيار پرهزینه است، چرا كه در اين روش ها مدل رياضی و يا نقشه كامل محيط مورد نیاز می باشد. اما تهیه مدل رياضی و يا نقشه كامل محيط کاری پيچيده است و با تغيير محيط ناگزير از به روز در آوردن مدل رياضی و يا نقشه آن خواهيم شد. باین ترتیب روش های سراسری برای حل چنین مسائلی غير عملی می باشند. پس به ناچار باید از روش های محلی استفاده نمود. در روش های محلی اطلاعات خام بصورت محلی در هر لحظه از زمان به ایجاد عملیات موتوری منجر می گردند. تهیه این اطلاعات بر عهده حسگرهای ربات متحرك می باشد.

سیستم های فازی ساده، به راحتی قابل درك و كم هزینه می باشند. این سیستم ها در موارد ساده ای كه قوانین فازی حاكم بر محيط را در دست داریم، به تنهایی جهت حل مساله قابل استفاده اند. در محيط های پيچيده تر استخراج و تنظیم قوانین فازی، جهت عملکرد مناسب و قابل قبول سیستم لازم می باشند.

در این تحقیق يك كنترل كننده فازی تطبیقی را معرفی نمودیم. در این كنترل كننده توابع عضویت فازی ورودی و خروجی، گausسی شكل انتخاب شده اند، اما پارامترهای آنها با توجه به شرایط محيط در هر قدم زمانی تعیین و تنظیم میگردد (كنترل كننده تطبیقی). تنظیم پارامترهای توابع عضویت با كمك اتوماتون یادگیر انجام می گیرد.

آزمایشات نشان می دهند كه اگر این پارامترها را در كل مسیر حرکت ثابت اختیار كنیم (كنترل كننده غيرتطبیقی)، مسیر حاصل ناهموارتر، طولانی تر، پرخطر تر و يا توام با تاخیر بیشتر خواهد شد. البته می توان مقادیر ثابتی برای این پارامترها یافت، كه منجر به پاسخ های قابل قبولی می گردند، اما این پاسخ ها به خوبی حالت تطبیقی نخواهند بود. یافتن این مقادیر نیز كار دشواری می باشد و هزینه محاسباتی و زمان زیادی لازم دارد. اتوماتون یادگیر در تعیین مقادیر مناسبی برای این پارامترها در شرایط مختلف محيط بسیار مطلوب عمل می نماید.

برای یافتن قوانین فازی سیستم در هر قدم زمانی، همانطور كه قبلاً اشاره شد، مستقیماً از

یادگیری تقویتی استفاده می نماییم. الگوریتم مورد استفاده، الگوریتم یادگیری Q می باشد. برای هر زوج حالت - عمل يك مقدار ارزش Q تعیین می گردد كه نشان می دهد هر عمل در هر حالت چقدر مناسب يا نا مناسب است. جهت رفتار همكاری از یادگیری Q با تابع ارزش توزیع شده استفاده می شود.

الگوریتم ما به دلیل عدم استفاده از رهبردهایی نظیر شبکه های عصبی یا الگوریتم های تكاملی كه در اغلب تحقیقات مورد استفاده قرار گرفته اند، ساده و روشن میباشد و چون نیازی به مشتق گیری و یاسایر عملیات پيچيده رياضی ندارد كم هزینه و سریع می باشد. ربات با كمك این كنترل كننده ساده میتواند مسیری نرم، هموار، كوتاه و حتی مناسب تر روش های مذکور را انتخاب كرده و به سمت هدف خود پیش رود. در این كنترل كننده ترکیب رفتارها به سادگی، باروشی قابل درك و با حداقل هزینه ممكن عملی میباشد. به كارگرفتن اتوماتون یادگیر باعث می شود كه كنترل كننده خود را با تغییرات محيط تطبیق دهد و در نتیجه با عوض شدن شرایط محيط عملکرد آن كاهش نمی یابد. نکته مهم این است كه تصمیم گیری با اطلاعات بسیار ناچیز در مورد طبیعت محيط صورت می گیرد و ربات بدون داشتن هرگونه پیش فرضی به تنهایی میتواند نحوه تصمیم گیری، انتخاب و اجرای عمل مناسب را بیاموزد.

در ادامه این پروژه می توان تحقیقات متنوع و مختلفی را انجام داد. به عنوان نمونه می توان با تغییر شكل و یا تعداد توابع عضویت فازی ورودی و خروجی، تاثیر این تغییرات در نحوه عملکرد سیستم را بررسی نمود و به روابط و یا نتایجی در این زمینه دست یافت. می توان بجای سیستم استنتاج فازی ممدانی از سیستم های استنتاج دیگری استفاده نمود و تاثیر این كار را بررسی كرد. می توان روش های دفازی سازی مختلف را آزمود. و یا این رفتارها را با تعداد سنسورهای بیشتر یا كمتر بررسی كرد. ما در الگوریتم یادگیری تقویتی خود، برای مشخص كردن پاداش یا تنبیه معیارهایی خاص در نظر گرفتیم و روابطی مشخص قرار دادیم. می توان برای هر رفتار از روابطی متفاوت استفاده كرد و الگوریتم را بررسی نمود و در صورت امکان به يك چارچوب رياضی برای تعیین این روابط دست یافت.

- [۷] Dong-Wook Lee, Kwee-Bo Sim, "An evolution of cellular automata neural systems using DNA coding method", *IEEE International Fuzzy Systems Conference Proceedings*, pp. 117_122, vol. 1, 1999.
- [۸] Cem Ünsal, "Intelligent Navigation of Autonomous Vehicles in an Automated Highway System: Learning Methods and Interacting Vehicles Approach", Dissertation submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering, Blacksburg, Virginia, January ۹, ۱۹۹۷.
<http://www.cs.cmu.edu/~unsal/diss/cemsdiss.html>
- [۹] Leslie Pack Kaelbling & Michael L. Littman, "Reinforcement Learning: A Survey",
<http://www.cs.brown.edu/people/lpk/rl-survey/rl-survey.html>
- [۱۰] <http://diwww.epfl.ch/lami/robots/k-amily/khepera.html>

برای رفتار همکاری می توان ظرافت های زیادتری قائل شد. مثلاً بجای نوبتی بودن توقف یک ربات، جهت عبور ربات دیگر، هر ربات پس از سنجش موقعیت خود و ربات دیگر تصمیم به حرکت یا توقف بگیرد. می توان به تحقیقاتی آماری جهت بررسی مساله پرداخت و به نتایج ارزشمندی دست یافت. یا برای تطبیقی کردن کنترل کننده از روشی بجز اتوماتون یادگیر استفاده نمود. یا با تغییر پارامترهای اتوماتون و یا یادگیری تقویتی به نتایجی جدید دست یافت.

مراجع

- [۱] Kai-Tai Song, Jen-Chau Tai, "Fuzzy Navigation Of A Mobile Robot", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 621_627, vol 1, 1992.
- [۲] Hee Rak Beom, Hyung Suck Cho, "A sensor-based navigation for a mobile robot using fuzzylogic and reinforcement learning", *IEEE Transactions on Systems, Man and Cybernetics*, pp. 464_477, vol 25, March 1995.
- [۳] Beom, H.R., Koh, K.C., Cho, H.S., "Behavioral control in mobile robot navigation using fuzzy decision making approach",
- [۴] *Robotic Systems and the Real IROS '94. Proceedings of the IEEE/RSJ/GI International Conference on intelligent Robots and Systems*, pp. 1938_1945, vol. 3, 1994.
- [۵] Yung, N.H.C., Ye, C., "Self-learning fuzzy navigation of mobile vehicle", *3rd International Conference on Signal Processing*, pp. 1465_1468 vol. 2, 1996.
- [۶] Fukuda, T., Kubota, N., "An intelligent robotic system based on a fuzzy approach", *Proceedings of the IEEE*, pp. 1448_1470, vol 87, Sept. 1999.