**یک الگوریتم بهینه‌سازی ازدحام ذرات تطبیقی مبتنی بر مساله برای تولید مسیر زاویه‌ای راه‌رفتن روبات دوپا**

معین اوحدی کارشک1 و محمدرضا اکبرزاده توتونچی2

1دانشجوی کارشناسی‌ارشد، قطب علمی رایانش نرم و پردازش هوشمند اطلاعات، دانشکده مهندسی، دانشگاه فردوسی مشهد

owhadi@stu.um.ac.ir

2استاد، قطب علمی رایانش نرم و پردازش هوشمند اطلاعات، دانشکده مهندسی، دانشگاه فردوسی مشهد

akbazar@um.ac.ir

چکیده - راه‌رفتن روبات دوپا به دلیل کوچک‌بودن ناحیه پشتیبان و نیز درگیربودن تعداد زیاد مفاصل بسیار پیچیده است. در گذشته، یافتن مسیر زاویه‌ای مفاصل روبات توسط سری شکسته فوریه، به عنوان یک روش مستقل از مدل، عملکرد مناسبی داشته‌است. در این مقاله، برای تقریب بهتر حرکت تناوبی راه‌رفتن، استفاده از سری کامل فوریه پیشنهاد شده‌است. هم‌چنین یک روش بهینه‌سازی تطبیقی جدید مبتنی بر الگوریتم بهینه‌سازی ازدحام ذرات برای یافتن مقادیر ضرائب سری فوریه کامل ارائه می‌شود. در روش پیشنهادی، استراتژی به‌روزرسانی سرعت ذرات با توجه به وضعیت جمعیت تغییر می‌کند و در ضمن، اطلاعات خاص مساله‌ی راه‌رفتن نیز در طراحی الگوریتم بهینه‌سازی مدنظر گرفته شده‌است. یافتن پاسخ بهینه قابل‌قبول و عدم همگرایی زودرس از ویژگی‌های الگوریتم ارائه‌شده است.

كليد واژه - الگوریتم بهینه‌سازی هوش ازدحامی تطبیقی، راه‌رفتن، روبات دوپا

# مقدمه

امروزه پژوهش بر روی روبات‌های انسان‌نما از موضوعات مورد علاقه‌ی محققان است. این روبات‌ها در تحقیقات مربوط به حرکت و جابه‌جایی به عنوان روبات‌های دوپا نیز شناخته می‌شوند. مطالعه بر روی روبات‌های دوپا از این نظر اهمیت دارد که اولاً، انسان‌ها در تعامل با آن‌ها احساس راحتی بیشتری می‌نمایند و بنابراین بیشتر به آن‌ها اعتماد کرده و تعامل بهتری با آن‌ها دارند و در ثانی، ساختار محیط زندگی انسان بر اساس آناتومی بدن اوست و در نتیجه، روباتی که به شکل انسان ساخته‌شود بهترین انطباق را با محیط انسانی خواهد داشت. توسعه‌ی یک روبات دوپا از جنبه‌های مختلفی مد نظر است که یکی از مهمترین آن‌ها راه رفتن است. توانایی راه‌رفتن خود شامل زیرمجموعه‌هایی همچون راه‌رفتن به سمت جلو، عقب و طرفین و همچنین چرخیدن درجا است. اصولاً جدای از ظاهر روبات‌های انسان نما، راه رفتن از ویژگی‌های بارز این‌گونه روبات‌ها است. راه‌رفتن بر روی دوپا حرکتی بسیار پیچیده است که بسیاری از مفاصل روبات در آن درگیر هستند. روش‌های مختلفی برای توسعه‌ی این توانایی پیشنهاد شده است که هدف تمام آن‌ها طراحی سیستمی برای راه رفتن روبات با سرعت و تعادلی مشابه راه رفتن انسان است. تلاش‌های انجام شده را در دو زیرگروه کلی می‌توان تقسیم نمود. راه‌رفتن ایستا ]1[ و راه‌رفتن پویا ]2.[

در راه‌رفتن ایستا، روبات به صورت استاتیک، پایدار فرض می‌شود. برای این منظور، مرکز جرم روبات همواره باید در ناحیه‌ی پشتیبان در زیر روبات قرار داشته‌باشد. محدودیت‌های این روش باعث شده است تا امروزه روش‌هایی بر پایه‌ی این نوع راه رفتن پیشرفت قابل توجهی نداشته باشند. راه رفتن پویا اما این امکان را ایجاد می‌کند تا برای لحظاتی محدود و کوتاه، مرکز جرم روبات از ناحیه‌ی پشتیبان خارج شود. این امر باعث می‌شود که راه رفتن پویا محدودیت‌های دینامیکی کمتری را نسبت به راه رفتن ایستا داشته‌باشد. هر چند که معیار خاصی برای تعیین میزان پایداری این نوع راه رفتن وجود دارد، می‌توان از نقطه‌ی گشتاور صفر[[1]](#footnote-1) به شرطی که مفصل مچ پای روبات قابلیت حرکت داشته و نیز همواره یک پای روبات بر روی زمین باشد، به عنوان معیاری برای پایداری بهره‌برد ]3[. در راه‌رفتن پویا دو رویکرد اصلی وجود دارد. برخی از روش‌ها با رویکرد مبتنی بر مدل هستند و برخی نیز مستقل از مدل طراحی می‌شوند.

در روش‌هایی که رویکردی مبتنی بر مدل دارند، اول باید مدل روبات ساخته شده و کنترل‌کننده‌ها بر مبنای آن مدل خاص طراحی می‌شوند. روش‌های کنترل ZMP ]4 [و پاندول معکوس‌شده ]5 [از موثرترین تلاش‌های انجام شده هستند که رویکردی بر مبنای مدل دارند. ZMP نقطه‌ای است که در آن برآیند نیروهای وارد بر روبات صفر است و عمل راه رفتن با کنترل این نقطه و اطمینان از قرار گرفتن آن در ناحیه‌ی پشتیبان صورت می‌گیرد. در روش پاندول معکوس‌شده، روبات دوپا به صورت یک جفت پاندول معکوس‌شده فرض و سپس کنترل می‌شود. اما در مقابل روش‌هایی که رویکردی بر مبنای مدل دارند، روش‌های دیگری نیز وجود دارند که کنترل روبات را مستقل از مدل آن انجام می‌دهند. در این روش‌ها، کنترل‌کننده بر اساس اطلاعات ورودی از سنسورها کنترل روبات را انجام می‌دهد و در نتیجه برای طراحی کنترل‌کننده نیازی به درنظر‌گرفتن یک مدل خاص از روبات نیست. پیاده سازی سایر توانایی‌های روبات در این روش‌ها آسان هستند. از تلاش‌هایی که بر اساس رویکرد مستقل از مدل هستند می‌توان به راه‌رفتن دینامیک غیرفعال[[2]](#footnote-2) ]6[، تولید‌کننده‌ی الگوی مرکزی ]7 [، روش پرتابه‌ای ]8[ و سری فوریه‌ی ناقص ]9 [اشاره کرد. در PDW روبات بدون عملگر فرض شده و مدل به گونه‌ای ساخته می‌شود تا با استفاده از جاذبه‌ی زمین حرکت نماید. محدودیت این روش آن است که تنها در سراشیبی قابل استفاده است. اما از طرف دیگر ترکیب این روش با سایر روش‌های پیشنهادی می‌تواند باعث کاهش مصرف انرژی شود. به گونه‌ای که اگر سطح زیرین روبات از شیب کافی برخوردار باشد از این روش و در شرایط محیطی دیگر از سایر روش‌های موجود استفاده می‌شود.CPG شبکه‌های عصبی مصنوعی هستند که از آن‌ها به عنوان تولیدکننده‌ی مسیر زاویه‌ای تناوبی استفاده می‌شود و مدل‌های ریاضی چون موارد ارائه‌شده در ]10 [و ]11 [در آن‌ها به کار می‌رود. در روشBallistic نیز از مدلی ساده شده از راه‌رفتن بهره گرفته شده‌است که در آن پای در حال حرکت، تنها در شروع و پایان طی کردن یک قدم، فعال و قابل کنترل درنظر گرفته می‌شود.

روش TFS روشی برای تولید مسیر زاویه‌ای است که عموماً از ZMP به عنوان معیار تعادل و پایداری بهره می‌برد. سادگی پیاده سازی و تعیین طول و فرکانس گام برداشتن بوسیله‌ی تنها یک پارامتر از مزایای این روش است. در این روش دو سیگنال برای بخش بالا و پایین سیگنال اصلی راه‌رفتن به طور مجزا مدل می‌شود و این باعث بروز اشکالاتی در این روش شده‌است. از جمله این که پیاده‌سازی سایر توانایی‌های مبتنی بر راه‌رفتن همچون چرخیدن و راه‌رفتن جهت دار در آن با مشکلاتی همراه است و همچنین تغییر وضعیت از وضعیت بخش بالا به پایین و بالعکس باعث ناپایداری و مصنوعی شدن حرکت می‌شود. دلیل استفاده از TFS و در نظر گرفتن دو بخش مجزا برای تولید سیگنال اصلی آن است که مدل کردن روبات با سری فوریه‌ی کامل نیازمند یافتن پارامتر‌های بیشتری است. جستجو در فضایی با ابعاد بالا نیز نیازمند استفاده از الگوریتم‌های جستجوی فرااکتشافی قدرتمند است.

در این مقاله، برای رفع مشکلات TFS از سری کامل فوریه استفاده شده،‌ هم‌چنین برای جستجو در فضای بزرگ‌تر یک الگوریتم بهینه‌سازی تطبیقی جدید ارائه شده‌است. اگرچه در این‌جا مسیر زاویه‌ای تنها برای ایجاد توانایی راه‌رفتن به سمت جلو انجام شده‌است، امکان پیاده‌سازی توانایی‌های متفاوت تنها با استفاده از تعریف یک تابع تناسب جدید میسر است. در تعریف تابع تناسب می‌توان به جز جهت حرکت، پارامترهایی چون مصرف انرژی، پایداری و مقاومت را نیز دخیل کرد. الگوریتم ازدحام ذرات در سال 1995 ارائه‌شد ]12[. این الگوریتم بر اساس حرکت ذرات در فضای جستجو عمل می‌کند. در این کار پژوهشی به‌روزرسانی سرعت هر ذره به شکل تطبیقی و با استفاده از اطلاعات مساله انجام شده‌است. در مساله‌ی راه‌رفتن، هر دسته از ضرائب فوریه که با استفاده از آن روبات بتواند به تعداد محدودی گام را بپیماید، یک جواب قابل‌قبول درنظر گرفته می‌شود. دلیل این موضوع آن است که در هر چند قدم نحوه‌ی تولید مسیر زاویه‌ای توسط برنامه‌ریز، که یک کنترل‌کننده‌ی سطح بالا است، اصلاح می‌گردد. در نتیجه، در الگوریتم بهینه‌سازی تطبیقی پیشنهادی، تمام پاسخ‌هایی که روبات با استفاده از آن‌ها در چند گام اولیه به زمین نمی‌افتد، به عنوان پاسخ بهینه سراسری فرض شده‌اند.

یک روش مبتنی بر بهینه‌سازی ازدحام ذرات برای استخراج پارامترهای حل تحلیلی در [13] ارائه گشته‌است. دیدگاه بهینه‌سازی چندهدفه نیز تا کنون به شکل موفقی عمل کرده‌است ]14] [15[ [16]. تا کنون از بهینه‌سازی ازدحام ذرات در تنظیم پارامترهای سایر روبات‌ها نیز استفاده شده‌است [17]. مقاله [18] نیز به معرفی یک روش جدید برای راه‌رفتن به کمک بهینه‌سازی ازدحام ذرات می‌پردازد. در [19] نیز یک مقایسه بین روش‌های فرااکتشافی مورد استفاده برای TFS انجام شده‌است.

در بخش بعدی نحوه‌ی انجام آزمایش‌ها و مدل روبات استفاده شده تشریح می‌شود. در بخش سه نیز نحوه‌ی تولید مسیر زاویه‌ای مورد بحث قرار می‌گیرد. الگوریتم بهینه‌سازی پیشنهادی در بخش چهار توضیح داده ‌شده‌است. نتایج این کار پژوهشی هم در بخش پنج آورده شده‌است. بخش شش نیز به نتیجه‌گیری و ارائه‌ی پیشنهادات برای کارهای آینده اختصاص یافته‌است.

# مدل روبات دوپا و شبیه‌سازی

از روبات شبیه‌سازی شده‌ی NAO ]20[ به عنوان مدل روبات دوپا استفاده شده‌است. دلیل استفاده از این مدل، استاندارد‌بودن و استفاده از آن در مسابقات روبوکاپ بوده‌است که امکان مقایسه‌ی روش پیشنهادی با سایر روش‌ها را افزایش می‌دهد. این روبات 5/4 کیلوگرم وزن و 57 سانتی‌متر قد و در مجموع 22 درجه آزادی دارد. شکل 1 آناتومی روبات NAO را نشان می‌دهد.

شکل 1 – آناتومی روبات NAO

بیشتر مفاصل روبات در راه‌رفتن یا در حفظ تعادل و کنترل روبات کاربرد دارند. اما برای جلوگیری از پیچیدگی کار و همچنین کاهش مصرف انرژی، برای راه‌رفتن مستقیم تنها مفاضل لگن، زانو و مچ که بیشترین تاثیر را داشتند، فعال و قابل‌حرکت در نظر گرفته‌شدند. برای کنترل هر مفصل هم از یک کنترل‌کننده‌ی PD [[3]](#footnote-3) استفاده شده‌است. در بخش بعد، نحوه‌ی ایجاد مسیر زاویه‌ای برای مفاصل درگیر روبات توضیح داده خواهد‌شد.

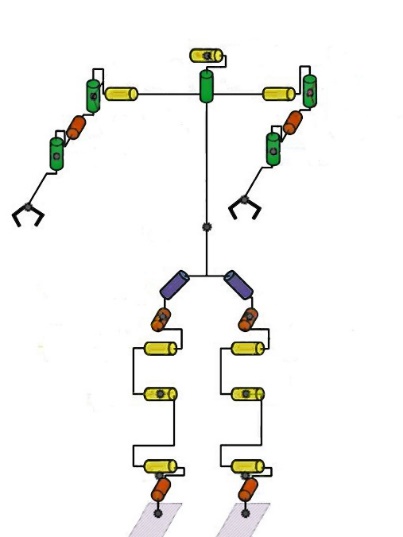
# تولید مسیر زاویه‌ای

راه رفتن حرکتی بسیار پیچیده است و از جهات گوناگونی می‌توان به مطالعه‌ی آن پرداخت. تولید مسیر زاویه‌ای و کنترل برای رسیدن به هدف، از رویکردهای موجود برای تحلیل توانایی راه‌رفتن هستند. تولید مسیر را به طور عمده می ‌توان به دو دسته‌ی تولید مسیر زاویه‌ای و مکانی تقسیم کرد. در تولید مسیر مکانی، هدف کنترل روبات از طریق مشخص‌کردن موقعیت مکانی عملگر نهایی روبات یعنی کف پاها در هر لحظه از زمان است. تولید مسیر زاویه‌ای نیز به این معنا است که مقدار زاویه‌ی هر مفصل در هر لحظه از زمان مشخص شود. در این کار پژوهشی هدف، تولید مسیر زاویه‌ای بهینه برای راه‌رفتن روبات دوپا به صورت پایدار است.

## 3-1- سری فوریه

در حالت کلی سری فوریه یک تابع ریاضی برای تولید و مدل‌کردن سیگنال‌های تناوبی به شکل معادله1 است. این معادله مقدار سیگنال را در لحظه‌ی مشخص می‌کند.

(1)

در این معادله مقدار بایاس سیگنال، فرکانس اصلی سیگنال و و ضرائب مجهول هستند. از نظر تئوری هنگامی که حد بالای سری فوریه بی‌نهایت در نظر گرفته‌شود، می‌توان هر سیگنال تناوبی را با معادله1 تخمین زد. اما استفاده از بی‌نهایت جمله در سری فوریه و همچنین یافتن ضرائب مجهول آن از نظر عملی امکان‌پذیر نمی‌باشد و در کاربرد‌های عملی این سری، از تعداد محدودی جمله استفاده می‌شود. در این مقاله تنها از سه جمله‌ی اول سری فوریه برای مدل‌کردن حرکات مفاصل استفاده شده‌است.

## 3-2- تولید مسیر زاویه‌ای با استفاده از سری فوریه

با توجه به تناوبی بودن حرکت راه رفتن بر روی دوپا و توانایی سری فوریه در مدل‌کردن این‌گونه توابع از سری فوریه برای تولید مسیر زاویه‌ای هر مفصل استفاده شده‌است ]21[. استفاده از سه جمله‌ی نخست سری فوریه، یافتن سه ضریب مجهول برای هر مفصل و در نتیجه 18 مجهول برای تمام شش مفصل مورد استفاده را ایجاب می‌کند. جستجو در فضای حالتی با این‌چنین ابعاد زیادی حتی برای الگوریتم‌های فرااکتشافی قدرتمند نیز کاری بسیار مشکل و زمان‌بر است.

با اتخاذ تدابیری می‌توان تعداد مجهولات را کم و فضای جستجو را کاهش داد. با تحلیلی بر مبنای ]22[ مشخص می‌شود که حرکت هر دو پا کاملاً مشابه هم بوده و تنها یک پا از پای دیگر تاخیری برابر نصف دوره‌ی تناوب دارد. بنابراین می‌توان با استفاده از تنها سه سری فوریه مسیر زاویه‌ای را برای یکی از دوپا محاسبه نمود و همان مسیر را با تاخیر ذکر‌شده به پای دیگر اعمال کرد. در نتیجه تعداد مسیرهای زاویه‌ای که باید تولید شوند به سه عدد و به تبع آن تعداد مجهولات به نه عدد کاهش می‌یابد. مسیر زاویه‌ای با استفاده از معادله‌های‌2 محاسبه شده‌است.

(2)

در این‌جا ، و مسیر زاویه‌ای مفاصل لگن، زانو و مچ در لحظه‌ی هستند. همچنین پارامترهای تا ضرائب مجهول و نیز فرکانس قدم‌برداشتن هستند که همان‌طور که در بخش4 تشریح خواهد شد، توسط الگوریتم بهینه‌سازی ازدحام ذرات تطبیقی مقادیر بهینه برای آن‌ها یافته خواهد شد.

# روش بهینه‌سازی

هوش جمعی نوعی روش جستجوی فرااکتشافی است که مبتنی بر رفتارهای جمعی در سامانه‌های نامتمرکز و خودسامانده بنیان شده است. الگوریتم‌های بهینه‌سازی مختلفی نیز بر مبنای همین مفهوم ایجاد شده‌است که الگوریتم بهینه‌سازی ازدحام ذرات ]12[ مثالی از آن است. سادگی و انعطاف‌پذیری این الگوریتم باعث شده‌است که تا کنون در مسائل مختلف مهندسی و ریاضی مورد استفاده قرار گیرد. الگوریتم ارائه شده‌ی اولیه ناپایدار بود و دچار همگرایی زودرس می‌شد و در نتیجه برای مسائل پیچیده مناسب نبود ]23[. به همین دلیل تا کنون نسخه‌های مختلفی از این الگوریتم ارائه شده‌است. ایجاد پایداری و تعریف درست مفهوم همسایگی از روندهایی است که بیشترین کارها در آن‌ها انجام شده‌است ]24[.

## 4-1- نسخه ی استاندارد الگوریتم

الگوریتم بهینه‌سازی ازدحام ذرات یک روش سراسری بهینه‌سازی است که با استفاده از آن می‌توان با مسائلی که جواب آن‌ها یک نقطه یا سطح در فضای چند بعدی می‌باشد، برخورد نمود. در این چنین فضایی، ذره‌هایی فرض و یک سرعت ابتدایی به آنها اختصاص داده می‌شود. همچنین همسایگی‌هایی بین ذرات لحاظ خواهدشد. سپس این ذرات در فضای پاسخ حرکت می‌کنند، و نتایج حاصله بر مبنای یک ملاک شایستگی پس از هر بازه‌ی زمانی محاسبه می‌شود. با گذشت زمان، ذرات به سمت نقاطی که دارای ملاک شایستگی بهتری هستند و در همسایگی یکسانی قرار دارند، شتاب می‌گیرند. در هر گام، سرعت جدید ذره با توجه به سرعت کنونی آن با استفاده از معادله3 محاسبه می‌شود.

*(3)*

در این‌جا سرعت جدید بعد ذره‌، اینرسی ذره، و ضرائب کشش به بهترین حالت محلی و سراسری، سرعت کنونی بعد ذره، و اعداد تصادفی در بازه‌ی صفر تا یک، مکان کنونی ذره و و نیز بهترین موقعیت محلی و سراسری دست‌یافته توسط مجموع ذرات است. جمله‌ی دوم که کشش به بهترین موقعیت محلی را نشان می‌دهد و جمله‌ی سوم که کشش ذره به بهترین موقعیت سراسری را نشان می‌دهد خوانده می‌شود. مکان جدید ذرات نیز با استفاده از معادله4 محاسبه می‌شود.

*(4)*

در این‌جا *مکان* *جدید ذره در بعد ،* مکان کنونی ذره در بعدو سرعت جدید بعد ذره‌ است. در نسخه‌ی استاندارد الگوریتم، میزان کشش به بهترین موقعیت محلی و سراسری با پارامترهای ، و تعیین می‌شود. همچنین مشخص خواهد کرد که ذره چقدر تمایل به تغییر سرعت داشته‌باشد. در نسخه‌ی کلاسیک همچنین همسایگی ذرات به شکل ایستا و ثابت فرض شده‌است. شکل2 کارکرد الگوریتم پایه را نشان می‌دهد.

شکل 2 – الگوریتم بهینه‌سازی ازدحام ذرات استاندارد

## 4-2- الگوریتم بهینه‌سازی ازدحام ذرات تطبیقی

در نسخه‌ی استاندارد الگوریتم، دقت کاوش را افزایش می‌دهد و از طرف دیگر باعث می‌شود تا جستجوی گسترده‌تری در فضای حالت انجام شود. با استفاده از هریک از این‌دو هر ذره می‌تواند هم نقش جستجوگر در محیط و هم نقش کاوش‌گر متمرکز جواب را داشته‌باشد. همچنین در یک گام خاص از اجرای الگوریتم، برخی از ذرات می‌توانند نقش کاوشگر در محیط، و برخی دیگر نقش یافتن پاسخ بهینه‌تر را بر عهده گیرند. مشکل اصلی در این بین، یافتن روش بهینه برای استفاده به‌موقع از هر یک از این دو رویکرد است. در الگوریتم پیشنهادی، از چهار روش مختلف برای بروزرسانی سرعت استفاده شده‌است. هر یک از این استراتژی‌ها برای شرایطی خاص در نظر گرفته‌ شده‌اند. این شرایط شامل وضعیت کل جمعیت و همسایگی ذره مورد‌نظر است. شکل3 استراتژی‌های موردنظر را نشان می‌دهد.

شکل 3 - بروزرسانی سرعت ذرات

*(5)*

*(6)*

*(7)*

در معادله7، تعداد ذرات جمعیت و وزن ذرات در جستجوی محیطی (در این‌جا مساوی برای تمام ذرات) را نشان می‌دهد.

چهار روش فوق در حقیقت، شیوه‌هایی مختلف از پیکربندی و تعامل بین ذرات را نشان می‌دهند. بروزرسانی نیز مانند ]25[ در سطح ذرات انجام خواهدشد. الگوریتم1 نحوه‌ی بروزرسانی سرعت را در الگوریتم پیشنهادی نشان می‌دهد.

1. **START**
2. Parameters: *acceptableAns*, *q*
3. Vlist := *Null*
4. Vlist.Add( getVal( 3 ) )
5. **if** *fitDiff* ≤ *ϵ* **then**
   1. Vlist.Add( getVal( 4 ) )
6. **end if**
7. **if** *pbestAve* ≤ *pbest* **then**
   1. Vlist.Add( getVal( 2 ) )
8. **end if**
9. **if** *acceptableAn* ≤ *gbest* **then**
   1. **if** Rand ≤ q **then**
      1. Vlist.Add( getVal( 1 ) )
   2. **end if**
10. *newVal* := Vlist.getItem();
11. **END**

الگوریتم1: بروزرسانی سرعت ذرات

**پارامترهای الگوریتم:**

* *acceptableAns: حداقل میزان قابل‌قبول تابع تناسب است. در مساله‌ی تولید مسیر زاویه‌ای برابر راه‌رفتن به اندازه‌ی 10 برابر طول ربات در نظر گرفته شده‌است.*
* *q: احتمال دنبال‌کردن در صورت رسیدن به پاسخ قابل‌قبول توسط جمعیت، که در این‌ مقاله 80 درصد فرض شده‌است.*

**ساختمان داده الگوریتم:**

*ساختمان داده Vlist یک آرایه برای ذخیره‌سازی مقادیر انتخابی سرعت است. تابع Add() یک مقدار را در این آرایه درج می‌کند. getItem() نیز یکی از مقادیر موجود در آرایه را به صورت تصادفی برمی‌گرداند. تابع getVal() نیز سرعت جدید را بر مبنای روش‌های ارائه‌شده در شکل2 محاسبه می‌کند. ورودی این تابع، شماره روش مورد نظر است.*

متغیرهای الگوریتم:

* *fitDiff:* میانگین تغییرات تابع تناسب در سه گام اخیر اجرا
* *ε:* حد آستانه‌ی تغییرات تابع تناسب (در این مقاله، حرکت کمتر از 0.1 متر روبات)
* *pbestAve:* میانگین بهترین پاسخ‌های محلی در کل جمعیت
* *pbest:* بهترین پاسخ در همسایگی
* *gbest:* بهترین پاسخ در کل جمعیت
* *newVal: سرعت جدید ذره*

## 4-3- تعریف توابع تناسب

تعریف صحیح تابع تناسب از مهمترین بخش‌های پیاده‌سازی الگوریتم بهینه‌سازی مورد بحث است. این تابع می‌بایست برای پاسخ‌های بهتر مقدار بیشتری را اخذ نماید. برای محاسبه‌ی مقدار تابع تناسب، ابتدا روبات در مبدا مختصات قرار می‌گیرد و سپس به مدت 100 ثانیه برای حرکت بر اساس مسیر زاویه‌ای تولید‌شده به آن مهلت داده می‌شود. همچنین چنان‌چه روبات بیافتد شبیه‌سازی خاتمه پیدا خواهد کرد. تابع برای حرکت رو به جلو به صورت معادله8 تعریف شده‌است.

(8)

در این‌جا مقدار تابع تناسب برای راه‌رفتن روبات به سمت جلو، تصویر نقطه‌ی نهایی حرکت روبات بر روی محور x و تصویر این نقطه بر روی محور y است. نیز نشان‌دهنده‌ی لحظه‌ی افتادن روبات خواهد‌بود. در بخش بعد روش پیشنهادی با استفاده از شبیه‌سازی آزمایش خواهد‌شد.

# شبیه‌سازی و نتایج

روش پیشنهادی برای پیاده‌سازی توانایی‌های پایه‌ی راه‌رفتن روبات دوپا بر روی سیستمی با پردازنده‌ی Intel Core2Due 2.66 GHz و حافظه‌ی 2GB و بر روی سیستم‌عامل Ubuntu 13.04 اجرا و الگوریتم بهینه‌سازی نیز با تعداد 50 ذره و 10 گام پیکربندی شده‌است.

در شکل3 نمودار تغییرات تابع تناسب در اجرای روش پیشنهادی برای راه‌رفتن مستقیم آورده شده‌است. در این نمودار الگوریتم پیشنهادی با نسخه‌ی استاندارد الگوریتم PSO و همچنین الگوریتم ژنتیک مقایسه شده‌است. پیکربندی نسخه‌ی استاندارد الگوریتم، مشابه پیکربندی الگوریتم پیشنهادی است. الگوریتم ژنتیک نیز دارای 50 فرد در هر نسل و با تکرار 10 عدد اجرا شده‌است. همچنین نرخ برش و جهش به ترتیب 0.85 و 0.04 بوده‌‍است. با توجه به دقیق بودن نسبی شبیه‌ساز فیزیکی مورد استفاده، دو پاسخ یکسان الزاما دو خروجی تناسب یکسان ندارند و بنابراین ممکن است که با وجود حفظ ذرات نخبه (در این مقاله، 10 درصد)، در برخی قدم‌ها شاهد کم‌شدن میزان تناسب بهترین ذره باشیم. با توجه به شکل4 الگوریتم پیشنهادی توانست مساله‌ی تولید مسیر زاویه‌ای را با 10 درصد بهبود نسبت به نسخه‌ی استاندارد الگوریتم حل نماید.

شکل 4 – تغییرات تابع تناسب در اجرای الگوریتم بهینه‌سازی

# نتیجه‌گیری و کارهای آینده

در این مقاله روشی برای تولید مسیر زاویه‌ای حرکات تناوبی راه‌رفتن معرفی شد. روش پیشنهادی اگرچه تعداد پارامترهای مجهول بیشتری نسبت به کارهای گذشته دارد، اما حرکتی متعادل‌تر و پایدارتر را فراهم می‌آورد. همچنین این روش می‌تواند برای تمام حرکاتی که بر مبنای راه‌رفتن هستند نیز مورد استفاده قرار گیرد. روش پیشنهادی برای بهبود الگوریتم ازدحام ذرات نیز توانست حل مساله را تا حد قابل قبولی بهبود بخشد. این بهبود از طریق تغییر استراتژی حرکت ذرات در شرایط مختلف ایجاد شده‌است. در آینده می توان با خوشه‌بندی پاسخ‌های مساله، مجموعه الگوها را برای تولید مسیر زاویه‌ای ایجاد نمود که در راه‌رفتن روبات در محیط واقعی بسیار مورد توجه است. همچنین با بهبود الگوریتم بهینه‌سازی می‌توان مفاصل بیشتری را به شکل فعال در نظرگرفتن و حرکت را پایدارتر نمود.

# مراجع

1. Kato I, Ohteru S, Kobayashi H, Shirai K, Uchiyama A, *“Information-power machine with senses and limbs”*, First CISM-IFToMM symposium on theory and practice of robots and manipulators, Springer-Verlag, 1974.
2. Takanishi A, Naito G, Ishida M, Kato I, *“Realization of plane walking by the biped walking robot WL-10R”*, Robotic and Manipulator Systems, 1982.
3. Hodgins JK, Raibert MH, *“Biped gymnastics”*, Int. J. Robot Res. 9(2), 1990.
4. M. Vukobratovic, B. Borovac, D. Surdilovic, *“Zero-moment point proper interpretation and new applications”*, In Proceedings of the 2nd IEEE-RAS International Conference on Humanoid Robots, Humanoids 2001, 2001.
5. S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, H. Hirukawa *“The 3D linear inverted pendulum mode A simple modeling for a biped walking pattern generation“*, In: Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001.
6. T. McGeer, *“Passive dynamic walking”,* International Journal of Robotics Research, Vol. 9(2), 1990.
7. C. Pinto, M. Golubitsky, *"Central Pattern Generator for Bipedal locomption"*, J. Math. Biol. ,Vol. 53, 2006.
8. S. Mochon and T.A. McMahon, *“Ballistic walking”,* Journal of Biomech, Vol. 13,1980.
9. L. Yang, C.M. Chew, A.N. Poo, *“Adjustable Bipedal Gait Generation using Genetic Algourithm Optimized fourier Series Formulation”*, In : Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006.
10. J. Buchli, F. Iida, A.J. Ijspeert, *"Finding Resonance: Adaptive Frequency scillators for Dynamic Legged Locomotion"*, Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006.
11. K. Matsuoka, “Sustained oscillations generated by mutually inhibiting neurons with adaptation”, Biol. Cybern., Vol. 52, 1985.
12. J. Kennedy and R.C. Eberhart, *“Particle Swarm Optimization”*, IEEE Int. Conf. on Neural Networks, 1995.
13. Huan, Tran Thien, Ho Pham Huy Anh, Y. Kondo, X. H. Han, X. Wei, Y. W. Chen, C. Wang et al. *“Novel Stable Walking for Humanoid Robot Using Particle Swarm Optimization Algorithm”*, hip 6, no. 6, 2015.
14. Mahmoodabadi, M. J., M. Taherkhorsandi, and A. Bagheri. “Optimal robust sliding mode tracking control of a biped robot based on ingenious multi-objective PSO”, Neurocomputing 124, 2014.
15. Taherkhorsandi, M., M. J. Mahmoodabadi, M. Talebipour, and K. K. Castillo-Villar. *“Pareto design of an adaptive robust hybrid of PID and sliding control for a biped robot via genetic algorithm optimization”*, Nonlinear Dynamics 79, no. 1, 2014.
16. Taherkhorsandi, M., K. K. Castillo-Villar, M. J. Mahmoodabadi, F. Janaghaei, and SM Mortazavi Yazdi*. “Optimal Sliding and Decoupled Sliding Mode Tracking Control by Multi-objective Particle Swarm Optimization and Genetic Algorithms”*, In Advances and Applications in Sliding Mode Control systems, pp. 43-78. Springer International Publishing, 2015.
17. Gao, Qin, Zhelong Wang, and Hongyi Li. *“An Optimization Algorithm with Novel RFA-PSO Cooperative Evolution: Applications to Parameter Decision of a Snake Robot”,* Mathematical Problems in Engineering, 2015.
18. Kherici, Nada, and Yamina Mohamed Ben Ali. *“Using PSO for a walk of a biped robot”*, Journal of Computational Science 5, no. 5, 2014.
19. Azarkaman, Mostafa, Mohammad Aghaabbasloo, and Mostafa E. Salehi. *“Evaluating GA and PSO evolutionary algorithms for humanoid walk pattern planning”*, In Electrical Engineering (ICEE), 2014 22nd Iranian Conference on, IEEE, 2014.
20. http://www.aldebaran-robotics.com/en/
21. Hinrichs RN, *“Whole body movement, coordination of arms and legs”*, In: Winters JM, WooSLY (eds) Walking and running. Springer-Verlag, New York, 1990.
22. L. Yang, C.M. Chew, T. Zielinska, A.N. Poo, *“A Uniform Bipedal Gait Generator with Offline Optimization and Online Adjustable Parameters”*, Robotica 2007, Cambridge University Press, United Kingdom, 2007.
23. J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baska, *“Comprehensive learning particle swarm optimizer for global optimization of multimodal functions”,* IEEE Trans. Evol. Comput., vol. 10, no. 3, 2006.
24. [K. KAMEYAMA](https://www.jstage.jst.go.jp/AF06S010ShsiKskGmnHyj?chshnmHkwtsh=Keisuke+KAMEYAMA) , *“Particle Swarm Optimization - A Survey”*, IEICE Transactions on Information and Systems, [Vol. E92.D](https://www.jstage.jst.go.jp/AF06S010SryTopHyj?sryCd=transinf&noVol=E92.D&noIssue=) [No. 7](https://www.jstage.jst.go.jp/AF06S010SryTopHyj?sryCd=transinf&noVol=E92.D&noIssue=7), 2009.
25. Changhe Li; Shengxiang Yang; Trung Thanh Nguyen, *"A Self-Learning Particle Swarm Optimizer for Global Optimization Problems",* Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on , vol.42, no.3, 2012.

1. Zero Moment Point (ZMP) [↑](#footnote-ref-1)
2. Passive Dynamic Walking (PDW) [↑](#footnote-ref-2)
3. Proportional – Derivative (PD) controller [↑](#footnote-ref-3)