

شبیه سازی و کنترل مسیر حرکت ربات متحرک همه جهته

مینا سادات ابراهیمی^۱

دانشجوی کارشناسی ارشد

مجید ساده دل^۲

استادیار

هدف اصلی مقاله دنبال کردن مسیر ربات همه جهته با استفاده از الگوریتم جهت یابی به روش کنترل رفتار مینا به منظور شبیه سازی نوین با جهت گیری دو ربات به سمت یکدیگر که همراه با عبور از موانع و تغییر وضعیت بین هر رفتار به نقطه‌ی هدف می‌رسد. با استفاده از سینماتیک ربات، روابط مربوط به سرعت و گشتاور چرخ‌ها بدست آمده است. ساختار سکوی ربات دارای سه مجموعه چرخ متعامد همه جهته می‌باشد که می‌تواند در همه جهات با قابلیت انتقال بدون دوران و انتقال توام با دوران به طور مستقل از هم، حرکت کند. به علاوه نتایج بدست آمده از شبیه سازی ارائه شده است و جهت حصول اطمینان و بررسی مزایای این روش کنترلی نسبت به روش‌های دیگر، با سایر مراجع صحت سنجی انجام گرفته است.

واژه های راهنما: کنترل مسیر ربات همه جهته متحرک، مدل سینماتیکی، الگوریتم جهت یابی، کنترل رفتار مینا

۱- مقدمه

در دهه های اخیر، ربات متحرک همه جهته^۳ توجه و تحقیقات روزافزون جوامع تحقیقاتی را به خود جلب کرده است. یکی از مزایای ربات‌های همه جهته با استفاده از چرخ های همه جهته^۴ این است که محدودیت غیر هولونومیکی ندارند که در سایر ربات‌های متحرک وجود دارد [۱]. با استفاده از سرعت دورانی هر چرخ همه جهته به عنوان ورودی، ربات می تواند به راحتی به هر کجا که کاربر فرمان بدهد، حرکت کند، که این قانون کنترلی ساده به راحتی قابل دست یافتن است. همانطور که در شکل (۱) نشان داده شده است، چرخ همه جهته از چرخ و غلتک تشکیل شده است، به این معنی که سرعت کل چرخ همه جهته ترکیبی از سرعت چرخ و سرعت غلتک است. کنترل ربات بسیار پیچیده است و گاهی اوقات لازم است محدودیت حالت ربات برای

^۱ دانشجوی کارشناسی ارشد، گروه مهندسی مکانیک، دانشگاه تربیت مدرس، ایران minasadat.ebrahimi@modares.ac.ir

^۲ نویسنده مسئول، استادیار، گروه مهندسی مکانیک، دانشگاه تربیت مدرس، ایران majid.sadedel@modares.ac.ir

تاریخ دریافت: ۱۴۰۰/۰۴/۰۲، تاریخ پذیرش: ۱۴۰۰/۰۷/۱۱

^۳ Omni-Directional Mobile Robot

^۴ Omni-wheel

تکمیل طراحی کنترل در نظر گرفته شود [۳ و ۲]. یک روش رایج به منظور کنترل یک ربات متحرک، طراحی کنترل حرکت فقط بر اساس معادله‌ی سینماتیکی ربات است [۴ و ۵]. به علاوه استفاده از مدل دینامیکی جهت کنترل حرکت ربات یک روش معمول است [۶-۸]. استفاده از این نوع ربات در مصارف خانگی بسیار رایج است، مقاله‌ی [۹] طراحی، پیاده سازی و ارزیابی یک مدل ربات دستیار شخصی^۱ سه چرخ (شکل ۲) را ارائه کرده است که در آن به بررسی سینماتیک سیستم حرکت و ارزیابی تخمین دنبال کردن مسیر از مقایسه‌ی جابجایی تخمینی با حرکت سنجی داخلی موتورها و فرآیند محلی کردن و نقشه برداری همزمان^۳ براساس اطلاعات سنسور لیدار^۴ پرداخته شده است. مقاله‌ی [۱۰] طراحی یک کنترل کننده‌ی تطبیقی مبتنی بر خطا با پاسخ سریع و مناسب برای یک کاربرد آنلاین به منظور کنترل ربات همه جهته با حرکت ربات چهار چرخ مکانوم^۵ را ارائه داده است. این کنترل کننده می‌تواند بدون عبور از مرزهای فضای کاری، با وجود خطاهای عملگر، ربات را در صورت وجود یا عدم وجود مانع در فضای کاری کنترل کند تا به مقصد برسد. مقاله‌ی [۱۱] یک روش مبتنی بر فازی به منظور جهت‌یابی یک ربات همه جهته‌ی سه چرخ را ارائه داده است که ربات مسیر را همزمان با دوری از موانع ثابت یا متحرک، به سمت هدف ثابت یا متحرک طی می‌کند، که یک کنترل کننده‌ی ساده با دو رفتار فازی "دنبال کردن هدف" و "دوری از موانع و دنبال کردن دیوار" به کار گرفته شده است. مقاله‌ی [۱۲] دنبال کردن مسیر یک ربات سه چرخ همه جهته با استفاده از تابع پتانسیل هارمونیک با کنترل پیش بینی مدل^۶ را بیان کرده است. مقاله‌ی [۱۳] از الگوریتم جهت‌یابی به روش کنترل رفتار مینا، جهت کنترل حرکت ربات برای حالت انتقال بدون دوران و انتقال همزمان با دوران به صورت مستقل از هم جهت عبور از موانع و رسیدن به هدف به کار برده است و با استفاده از روابط سینماتیکی، سرعت دورانی و خطی چرخ‌ها بدست آمده است [۱۴]. به علاوه از روابط دینامیکی ربات جهت کنترل سرعت و جهت‌گیری ربات استفاده شده است [۱۵]، به عبارتی گشتاور سه موتور به عنوان ورودی به ربات اعمال می‌شود تا حرکت مطلوب حاصل شود [۱۶]. مقاله‌ی [۱۷] یک روش میدان پتانسیل نوین برای مسیریابی ربات متحرک در یک محیط ثابت که هدف و موانع آن ثابت هستند، پیشنهاد داده است، هنگامی که میدان پتانسیل ایجاد شد، مسیریابی بسیار سریع پیش می‌رود زیرا فقط شامل محاسبه‌ی عوامل گرانش می‌باشد. مسیر ایجاد شده نیز دارای ویژگی‌های مطلوب مختلف مانند آزادی از حداقل موضعی^۷ بدون مشکل لرزیدن، نرمی و دوری از برخورد می‌باشد. به علاوه مقاله‌ی [۱۸] یک روش میدان پتانسیل برای دنبال کردن مسیر و دوری از موانع برای بازوهای اضافی ارائه داده است، که نه تنها بر موقعیت بازوی کارگیر متمرکز شده است بلکه وضعیت آنها را در طول دنبال کردن مسیر و دوری از موانع در نظر گرفته است. مقاله‌ی [۱۹] به منظور مطالعه‌ی سازگاری ربات مار مانند در حرکات زیر آبی، یک الگوریتم کنترل دو بعدی دوری از موانع برای یک ربات مار مانند بر اساس غوطه وری

¹ Assistant Personal Robot (APR)

² Odometry

³ SLAM procedure

⁴ LIDAR sensor

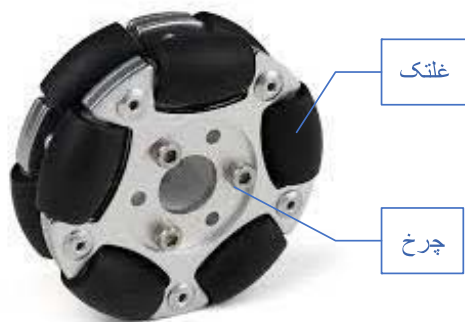
⁵ Mecanum

⁶ Model Predictive Control (MPC)

⁷ Local Minima

روش بولتزمن شبکه‌ای مرزی^۱ و میدان پتانسیل مصنوعی بهبود یافته^۲ ارائه داده است، که دوری از موانع ربات در یک میدان جریان با بهینه سازی معادله‌ی خمیدگی منحنی مار و حذف حداقل موضعی در روش میدان پتانسیل محقق شده است. به طور کلی مهم‌ترین مزیت این روش نسبت به روش‌هایی مانند میدان پتانسیل گیر نکردن ربات در حداقل موضعی می‌باشد و با وجود موانع با اشکال پیچیده و نزدیک به هم این روش به درستی عمل می‌کند.

این مقاله به این صورت بخش بندی شده است: بخش دوم مدل سینماتیکی و دینامیکی ربات مورد بررسی قرار گرفته است، در بخش سوم یک سیستم کنترل با عملکرد بالا به منظور دنبال کردن مسیر ربات، ارائه شده است. در بخش چهارم الگوریتم جهت‌یابی به روش کنترل رفتار مینا معرفی شده است. در بخش پنجم صحت این روش کنترلی در مقایسه با سایر مراجع سنجیده شده است. در بخش ششم دو حالت شبیه سازی و نتایج مربوط به آنها بررسی شده است.



شکل ۱- چرخ همه جهته



شکل ۲- ربات دستیار شخصی [۹]

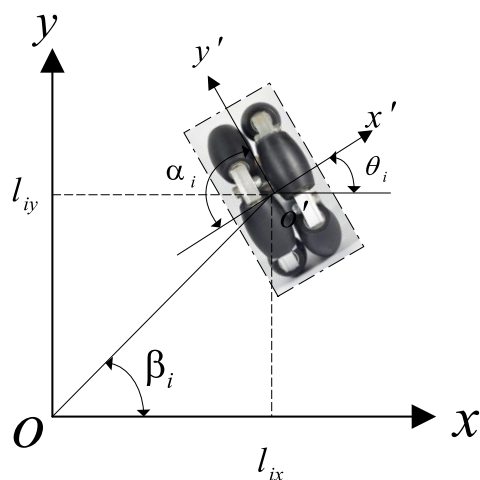
¹ Immersed boundary-lattice Boltzmann method (IB-LBM)

² Improved artificial field (APF)

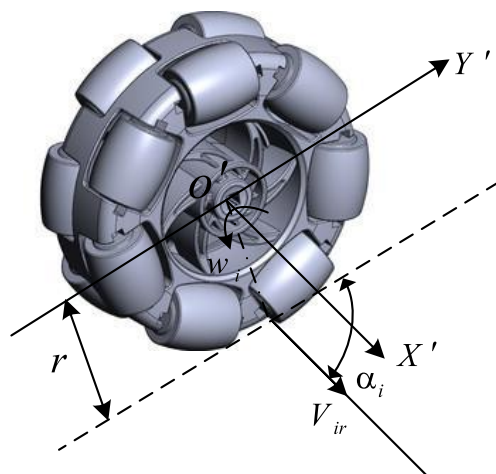
۲- مدلسازی ربات همه جهته

۲-۱- مدل سینماتیکی

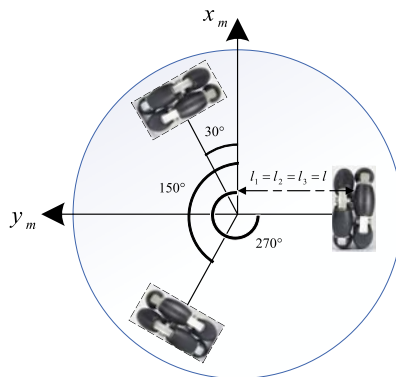
برای ربات همه جهته‌ی مورد نظر، هر چرخ دارای دو پارامتر موقعیت می‌باشد: موقعیت نسبت به مرکز سکوی ربات و زاویه‌ی مربوطه است. مختصات چرخ همه جهته در شکل (۳) نشان داده شده است. در این شکل، xOy ، دستگاه مختصات متصل به مرکز سکوی ربات است و $x'Oy'$ دستگاه مختصات متصل به مرکز چرخ است که در اینجا 90° درجه می‌باشد، β_i زاویه‌ی بین خط‌های \overline{Ox} و $\overline{Ox'}$ می‌باشد و $(l_{ix}, l_{iy}, \theta_i)$ موقعیت چرخ (در دستگاه مختصات، xOy) می‌باشد. ساختار چرخ همه جهته در شکل (۴) نشان داده شده است. در این شکل شعاع هر چرخ، r بردار سرعت مرکز هر غلتک و w_i سرعت دورانی موتور است. بردار سرعت $[v_x \ v_y \ w]^T$ سرعت مرکز سکوی ربات را نسبت به زمین تعریف می‌کند و $[v_{ix} \ v_{iy} \ w_i]^T$ سرعت مرکز هر چرخ را نسبت به مختصات xOy بیان می‌کند و $[v'_{ix} \ v'_{iy} \ w'_i]^T$ سرعت مرکز هر چرخ را نسبت به مختصات $x'Oy'$ تعریف می‌کند.



شکل ۳- موقعیت و حالت چرخ در مرکز سکو



شکل ۴- موقعیت و حالت چرخ در مرکز سکو



شکل ۵- مدل ربات متحرک همه جهته

جدول ۱- پارامترهای ساختار سیستم

| پارامتر | β_1 و θ_1 | β_2 و θ_2 | β_3 و θ_3 | α_1 | α_2 | α_3 |
|---------|------------------------|------------------------|------------------------|------------|------------|------------|
| مقدار | 30° | 150° | 270° | 90° | 90° | 90° |

با در نظر گرفتن هندسه‌ی ربات مورد نظر پارامترهای ربات را استخراج می‌کنیم (شکل ۵). مدل سینماتیکی ربات با توجه به مرجع [۱۴] به صورت زیر حاصل می‌شود.

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = -\frac{1}{r} \begin{bmatrix} \frac{1}{2} & -\cos 30 & -l \\ \frac{1}{2} & \cos 30 & -l \\ -1 & 0 & -l \end{bmatrix} \begin{bmatrix} v \cos \phi \\ v \sin \phi \\ w \end{bmatrix} \quad (1)$$

l در رابطه‌ی بالا فاصله‌ی بین هر چرخ و مرکز ثقل ربات است.

۲-۲- مدل دینامیکی

فرض می‌کنیم که سیستم مختصات اصلی $x_w o_w y_w$ بر روی زمین ثابت است و سیستم مختصات $x_m o_m y_m$ متحرک بر روی مرکز ثقل ربات ثابت است (شکل ۶).

با استفاده از روابط [۲۰]، دینامیک برای ربات متحرک به صورت زیر بدست می‌آید:

$$M(\ddot{x}_m - \dot{y}_m \dot{\phi}) = f_x \quad (2)$$

$$M(\ddot{y}_m - \dot{x}_m \dot{\phi}) = f_y \quad (3)$$

$$I_v \ddot{\phi} = M_I \quad (4)$$

به علاوه مشخصه‌ی سیستم حرکتی [۲۱] برای هر چرخ به صورت زیر می‌باشد:

$$I_w \dot{w}_i + c w_i = k u_i - r D_i \quad (5)$$

$$(i = 1, 2, 3)$$

که l فاصله‌ی بین هر چرخ و مرکز ثقل ربات است، c ثابت اصطکاک ویسکوز هر چرخ، D_i نیروی محرکه برای هر چرخ است، r شعاع چرخ، I_w ممان اینرسی چرخ حول محور آن، w_i سرعت دورانی چرخ، k ثابت راندن و u_i گشتاور ورودی می‌باشد. از طرفی، رابطه‌ی هندسی بین متغیرهای $\dot{\phi}$ ، \dot{x}_m و \dot{y}_m یا همان سینماتیک معکوس به صورت رابطه‌ی (۱) می‌باشد، بنابراین با استفاده از روابط ذکر شده در [۱۵] معادلات زیر استخراج می‌شوند:

$$\ddot{x}_m = a_1 \dot{x}_m + \dot{a}_2 \dot{y}_m \dot{\phi} - b_1(u_1 + u_2 - 2u_3) \quad (۶)$$

$$\ddot{y}_m = a_1 \dot{y}_m + \dot{a}_2 \dot{y}_m \dot{\phi} + \sqrt{3}b_1(u_1 - u_2) \quad (۷)$$

$$\ddot{\phi} = a_3 \dot{\phi} + b_2(u_1 + u_2 + u_3) \quad (۸)$$

تعریف متغیرهای حالت برای ربات به صورت $x = [x_m \quad y_m \quad \phi \quad \dot{x}_m \quad \dot{y}_m \quad \dot{\phi}]^T$ می‌باشد، ورودی‌ها به صورت $u = [u_1 \quad u_2 \quad u_3]^T$ و متغیرهای خروجی به صورت $y = [\phi \quad \dot{x}_m \quad \dot{y}_m]^T$ می‌باشند و در ادامه فرم فضای حالت به شکل زیر حاصل می‌شود:

$$\dot{x} = A(x)x + B(x)u \quad (۹)$$

$$y = Cx \quad (۱۰)$$

که؛

$$A(x) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_1 & -a_2 \dot{\phi} & 0 \\ 0 & 0 & 0 & a_2 \dot{\phi} & a_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_3 \end{bmatrix}$$

$$B(x) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ b_1 \beta_1 & b_1 \beta_2 & 2b_1 \cos \phi \\ b_1 \beta_3 & b_1 \beta_4 & 2b_1 \sin \phi \\ b_2 & b_2 & b_2 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

اطلاعات جدول بالا مربوط به مرجع [۱۵] می‌باشد.

۳- کنترل ربات همه جهته

از آنجایی که معادله‌ی دینامیکی در سیستم مختصات مطلق در نظر گرفته شده است، با حل معادله‌ی (۹) با توجه به u_i ، $i = 1, 2, 3$ ، کنترل شتاب حل شده به صورت ذیل حاصل می‌شود:

$$u_1 = \frac{\beta_1(\ddot{x}_w^* - a_1\dot{x}_w + a_2\dot{\phi}\dot{y}_w)}{6b_1} + \frac{\beta_1(\ddot{y}_w^* - a_1\dot{y}_w + a_2\dot{\phi}\dot{x}_w)}{6b_1} + \frac{\ddot{\phi}^* - a_3\dot{\phi}}{3b_2} \quad (11)$$

$$u_2 = \frac{\beta_2(\ddot{x}_w^* - a_1\dot{x}_w + a_2\dot{\phi}\dot{y}_w)}{6b_1} + \frac{\beta_4(\ddot{y}_w^* - a_1\dot{y}_w + a_2\dot{\phi}\dot{x}_w)}{6b_1} + \frac{\ddot{\phi}^* - a_3\dot{\phi}}{3b_2} \quad (12)$$

$$u_3 = \frac{\cos\phi(\ddot{x}_w^* - a_1\dot{x}_w + a_2\dot{\phi}\dot{y}_w)}{3b_1} + \frac{\sin\phi(\ddot{y}_w^* - a_1\dot{y}_w + a_2\dot{\phi}\dot{x}_w)}{3b_1} + \frac{\ddot{\phi}^* - a_3\dot{\phi}}{3b_2} \quad (13)$$

که \ddot{x}_w^* ، \ddot{y}_w^* و $\ddot{\phi}^*$ به ترتیب شتاب خطی در جهت x ، شتاب خطی در جهت y و شتاب زاویه ای به دست آمده از طریق اضافه کردن کنترل کننده‌ی PI به \ddot{x}_{wd} ، PI به \ddot{y}_{wd} و PD به $\ddot{\phi}_d$ می‌باشند به طوری که؛

$$\ddot{x}_w^* = \ddot{x}_{wd} + K_{\dot{x}P}e_{\dot{x}} + K_{\dot{x}I} \int_0^t e_{\dot{x}} dt \quad (14)$$

$$\ddot{y}_w^* = \ddot{y}_{wd} + K_{\dot{y}P}e_{\dot{y}} + K_{\dot{y}I} \int_0^t e_{\dot{y}} dt \quad (15)$$

$$\ddot{\phi}^* = \ddot{\phi}_d + K_{\phi P}e_{\phi} + K_{\phi D}\dot{e}_{\phi} \quad (16)$$

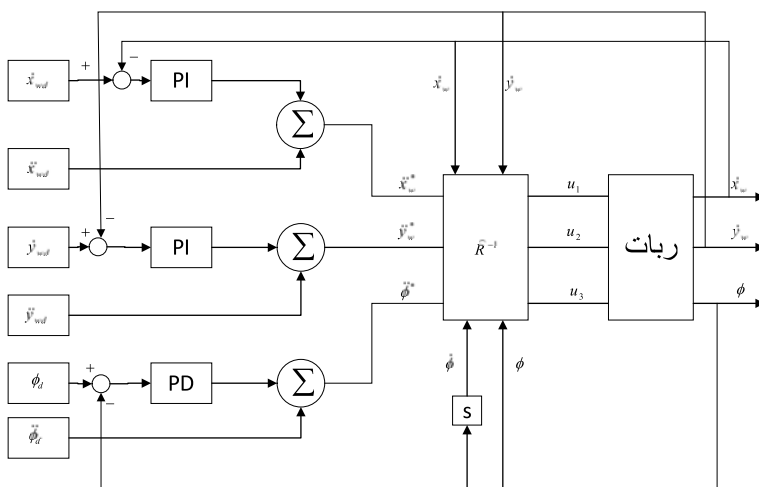
در اینجا K_P بهره‌ی تناسبی، K_I بهره‌ی انتگرال گیر و K_D بهره‌ی مشتق گیر می‌باشند و هر خطا به صورت زیر تعریف می‌شود:

$$e_{\dot{x}} = \dot{x}_{wd} - \dot{x}_w \quad (17)$$

$$e_{\dot{y}} = \dot{y}_{wd} - \dot{y}_w \quad (18)$$

$$e_{\phi} = \phi_d - \phi \quad (19)$$

نمودار بلوکی کنترل ربات متحرک در شکل (۶) نشان داده شده است، که در آن \hat{R}^{-1} مدل دینامیک معکوس تخمینی می‌باشد.



شکل ۶- شماتیک کنترل PID

۴- الگوریتم جهت یابی به روش کنترل رفتار مبنا^۱

در این مقاله از الگوریتم جهت یابی با هدف رسیدن به موقعیت مطلوب و دوری از موانع در دو حالت انتقال بدون دوران و انتقال توام با دوران به صورت مستقل از هم استفاده شده است، که در نهایت با داشتن گشتاور هر چرخ و استفاده از آنها به عنوان ورودی به ربات در کنترل کننده‌ی PID، می‌توان میزان دقت دنباله روی سرعت های خطی و زاویه ای ربات و همچنین موقعیت ربات را بررسی کرد که در بخش (۳-۶) نتایج آنها جهت صحت سنجی گشتاورهای بدست آمده از کنترل رفتار مبنا مورد بررسی قرار می‌گیرد و همچنین مزیت استفاده از کنترل کننده‌ی PID نسبت به کنترل کننده‌ی فیدبک حالت آدر کاهش خطاها نشان داده می‌شود. در این روش کنترلی می‌توان رفتار دنبال کردن دیوار^۲ را به صورت مود لغزشی^۳ بین رفتار رفتن به هدف^۴ و دوری از مانع^۵ نمایش داد. ورودی همان سرعت ربات است که به صورت $u = [\dot{x} \quad \dot{y}]^T$ در نظر گرفته می‌شود و برای رسیدن به هدف، دوری از موانع هنگام نزدیک‌تر شدن از اندازه‌ی معین و دنبال کردن دیوار جهت عبور از موانع بدست می‌آید:

تعیین سرعت ربات جهت رسیدن به هدف:

$$e_G = x_G - x \quad (20)$$

$$K_{GTG} = \frac{v_0(1 - e^{-\alpha\|e\|^2})}{\|e\|} \quad (21)$$

$$u_{GTG} = K_{GTG} * e_G \quad (22)$$

تعیین سرعت ربات جهت دوری از موانع:

$$e_O = x_O - x \quad (23)$$

$$K_{AO} = \frac{1}{\|e\|} \left(\frac{c}{\|e\|^2 + \varepsilon} \right) \quad (24)$$

$$u_{AO} = -K_{AO} * e_O \quad (25)$$

تعیین سرعت ربات جهت دنبال کردن دیوار به صورت ساعتگرد و پادساعتگرد:

$$u_{FW}^c = \beta R \left(\frac{-\pi}{2} \right) u_{AO} = \beta \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} u_{AO} \quad (26)$$

$$u_{FW}^{cc} = \beta R \left(\frac{\pi}{2} \right) u_{AO} = \beta \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} u_{AO} \quad (27)$$

¹ Behavior-based control

² State feedback controller

³ Follow wall

⁴ Sliding mode

⁵ Go to goal

⁶ Avoid obstacle

می توان با سعی و خطا با تنظیم پارامترهای v_0 ، α ، c ، ε و β ، ضرایب K_{GTG} و K_{AO} را به طور مناسب تعیین کرد.

۵- صحت سنجی

۵-۱- صحت سنجی حرکت دایره ای با جهت گیری ثابت

ابتدا از روش کنترل رفتار مبنا جهت دنبال کردن مسیر دایره ای نشان داده شده در شکل (۷) استفاده شده است [۱۶]. پس از بدست آوردن گشتاورهای هر چرخ از روابط بخش (۲)، ولتاژهای موتور بدست آمده، با مرجع (شکل ۸) مقایسه شده است. سپس با استفاده از دینامیک ربات میزان دقت دنبال کردن سرعت خطی ربات از طریق کنترل کننده ی PID (شکل های ۱۰ و ۱۱) مورد بررسی قرار گرفته است. شروع حرکت ربات در نقطه ی (۵، ۵) می باشد، به علاوه زاویه ی سر ربات در تمام طول مسیر صفر درجه می باشد (شکل ۹).

پارامترهای موجود در روابط ۲۰-۲۷ برای شبهه سازی به صورت زیر در نظر گرفته شده است:

$$v_0 = 10, \alpha = 2$$

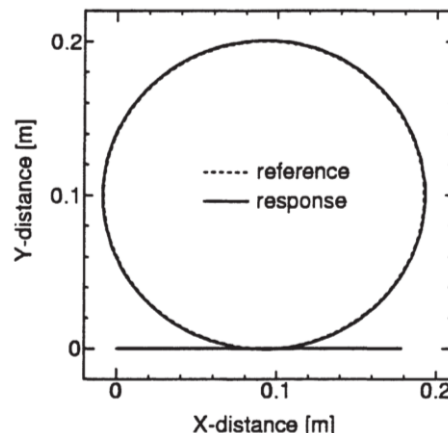
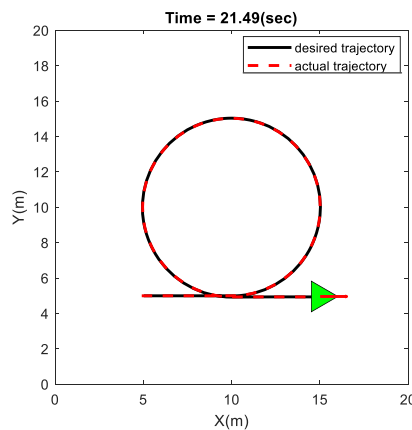
ضرایب کنترل کننده ی PID به صورت زیر در نظر گرفته شده است:

$$K_{\dot{x}P} = 300, K_{\dot{x}I} = 150$$

$$K_{\dot{y}P} = 300, K_{\dot{y}I} = 200$$

$$K_{\theta P} = 2, K_{\theta D} = 10$$

انتخاب v_0 و α در نهایت مقدار ضریب K_{GTG} را تعیین می کنند، افزایش این دو ضریب باعث سریعتر قرار گرفتن ربات در مسیر مطلوب می شود و بالعکس. مقدار ضرایب باید به گونه ای انتخاب شوند که ربات به خوبی مسیر را دنبال کند و اگر کمتر از مقدار تعیین شده باشند باعث می شود که ربات در مسیر مطلوب قرار نگیرد یا با خطای زیاد در مسیر قرار بگیرد، در نهایت با سعی و خطا مقادیر بالا برای این دو ضریب تعیین شده است.



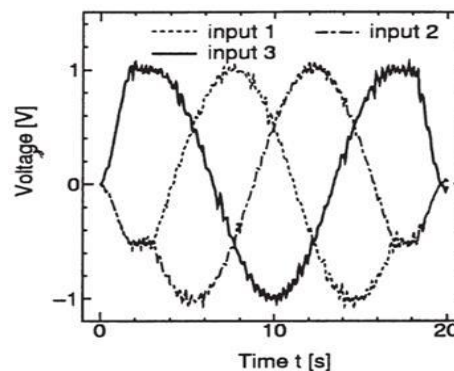
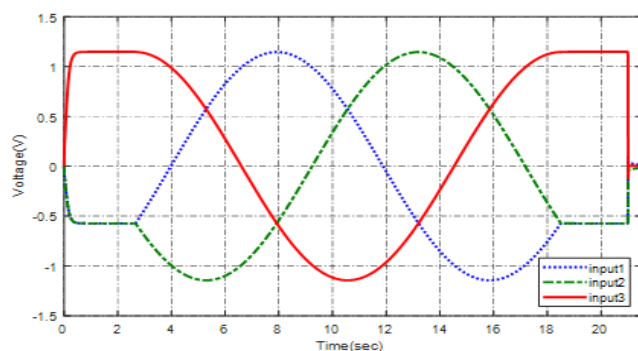
شکل ۷- مسیر دایره ای شکل (شکل سمت راست مربوط به مرجع [۱۶] می باشد).

پس از جهت یابی ربات با روش کنترل رفتار مبنا، ولتاژ موتورها به عنوان ورودی جهت کنترل موقعیت و سرعت ربات با استفاده از کنترل کننده PID، به ربات داده می‌شوند. همان طور که در شکل (۸) مشاهده می‌شود نتایج کاملاً شبیه به هم هستند در نتیجه این روش مسیر را بسیار مناسب دنبال کرده است که این صحت روابط و روش کنترلی را نشان می‌دهد. در ادامه از ورودی‌ها در کنترل کننده PID جهت دنبال کردن خروجی‌ها $y = [\phi \quad \dot{x}_m \quad \dot{y}_m]^T$ استفاده شده است که نتایج بدست آمده در این مقاله با نتایج مرجع ذکر شده در شکل‌های (۹-۱۱) مقایسه شده است و نتایج بسیار نزدیک به مرجع می‌باشد. شکل (۹) مربوط به دنبال کردن زاویه‌ی سر ربات می‌باشد که با خطای بسیار کمتری نسبت به مرجع زاویه‌ی مطلوب را دنبال می‌کند که این بهبود می‌تواند به دلیل ضرایب کنترلی انتخاب شده باشد. شکل (۱۰) سرعت در جهت x را نشان می‌دهد که نسبت به مرجع بسیار بهتر عمل کرده است که دلیل آن می‌تواند اضافه کردن ضریب D جهت کنترل سرعت در جهت x باشد، به علاوه مقدار سرعت مطلوب 0.4488 متر در ثانیه در نظر گرفته شده است. همچنین شکل (۱۱) مربوط به سرعت در جهت y می‌باشد که به دلیل ذکر شده دارای خطای کمتری نسبت به مرجع می‌باشد. ابتدا از روش کنترل رفتار مبنا جهت دنبال کردن مسیر خطی نشان داده شده در شکل (۱۲) استفاده شده است [۱۴]. پس از بدست آوردن سرعت خطی چرخ‌ها از روابط بخش (۲)، سرعت‌های بدست آمده با مرجع (شکل ۱۵) مقایسه شده است. در مرجع ذکر شده از سرعت چرخ‌ها جهت رسیدن به هدف استفاده شده است، در این مقاله از این روش کنترلی جهت دنبال کردن نقطه‌ی مورد نظر استفاده شده است و سرعت خطی چرخ‌ها به عنوان خروجی بدست آمده است و سپس با استفاده از دینامیک ربات و سرعت چرخ‌های بدست آمده به عنوان ورودی می‌توان میزان دقت دنبال کردن سرعت و موقعیت ربات از طریق کنترل کننده PID را مورد بررسی قرار داد. شروع حرکت ربات در نقطه‌ی $(0, 0)$ و نقطه‌ی هدف در $(2, 3)$ می‌باشد، به علاوه زاویه‌ی اولیه‌ی سر ربات صفر درجه $(\phi_0 = 0)$ و زاویه‌ی مطلوب 60 درجه $(\phi_d = \frac{\pi}{3})$ می‌باشد. پارامترهای موجود در روابط ۲۰-۲۷ برای شبیه سازی به صورت زیر در نظر گرفته شده است:

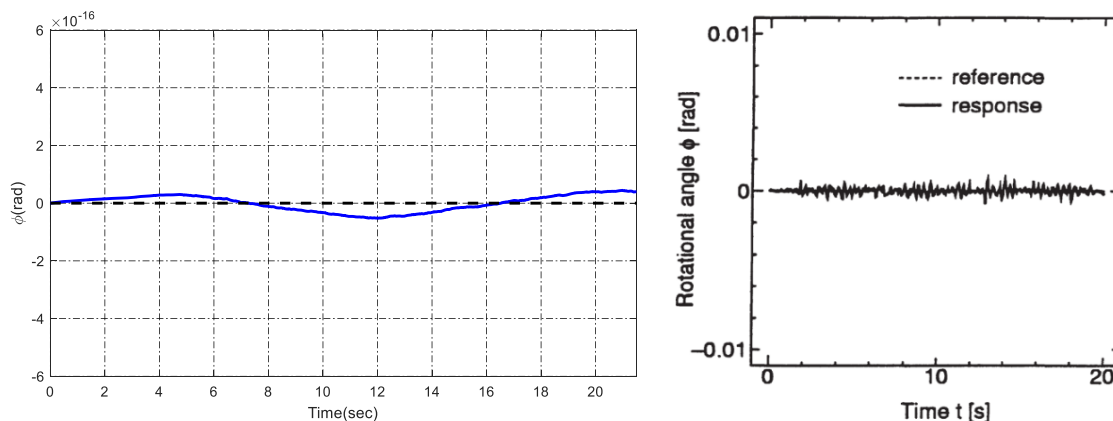
$$v_0 = 8, \alpha = 0.8$$

به علاوه با استفاده از ضریب P زاویه‌ی سر ربات کنترل می‌شود که متناسب با سرعت خطی ربات ضریب مناسب به صورت زیر انتخاب شده است:

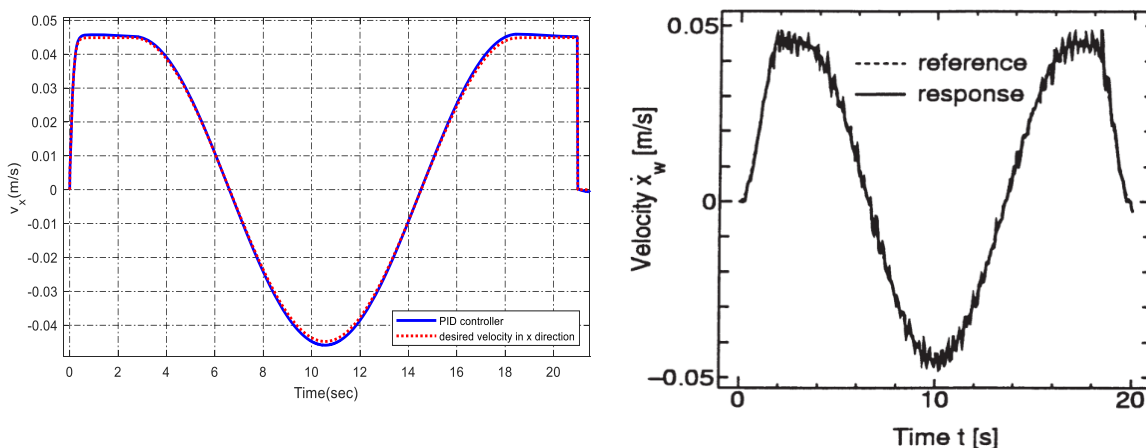
$$w = 5 * \text{atan2}(\sin(\phi_d - \phi), \cos(\phi_d - \phi))$$



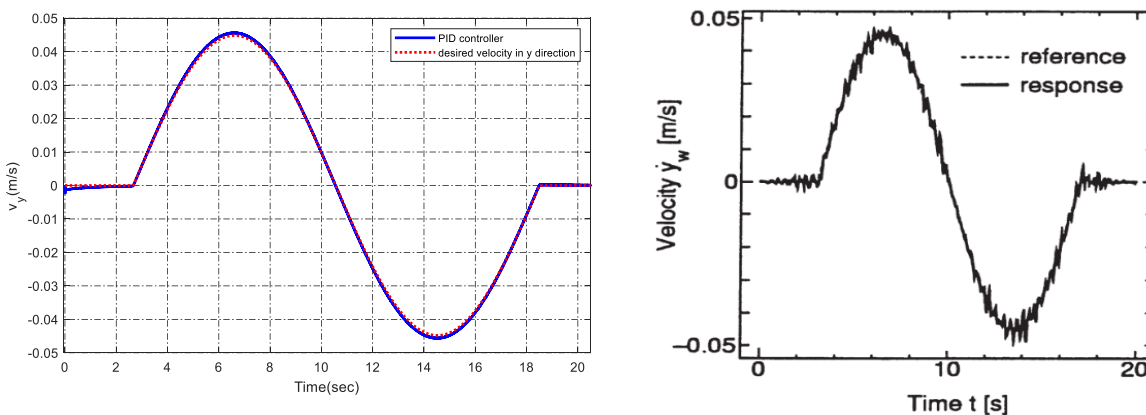
شکل ۸- ولتاژ ورودی برای مسیر دایره‌ای (شکل سمت راست مربوط به مرجع [۱۶] می‌باشد).



شکل ۹- زاویهی سر ربات برای حرکت دایره‌ای (شکل سمت راست مربوط به مرجع [۱۶] می‌باشد).



شکل ۱۰- سرعت در جهت x برای مسیر دایره‌ای (شکل سمت راست مربوط به مرجع [۱۶] می‌باشد).

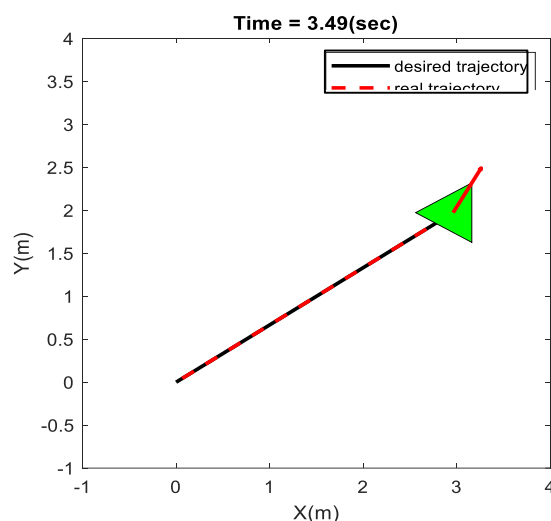


شکل ۱۱- سرعت در جهت y برای مسیر دایره‌ای (شکل سمت راست مربوط به مرجع [۱۶] می‌باشد).

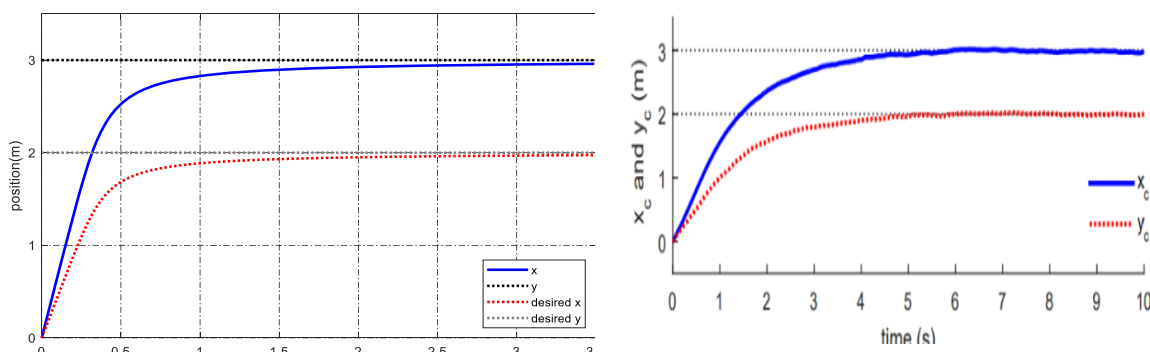
۲-۵- صحت سنجی حرکت ربات به سمت نقطه‌ی هدف

همان طور که در شکل (۱۳) مشاهده می‌شود این کنترل کننده به خوبی به نقطه‌ی هدف همگرا شده است و نتیجه‌ی آن بسیار به مرجع ذکر شده نزدیک می‌باشد و همچنین زاویهی مطلوب نیز به خوبی کنترل شده

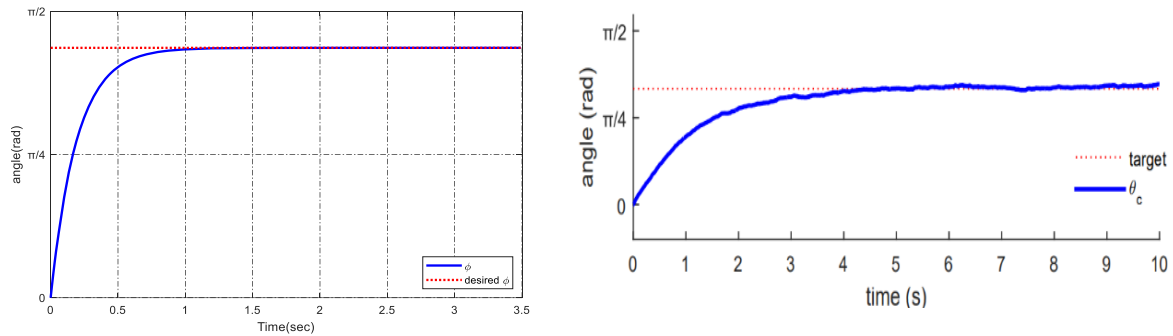
است (شکل ۱۴). نتیجه‌ی حاصل در شکل (۱۳) نشان می‌دهد که ربات در ثانیه‌ی سوم به نقطه‌ی هدف همگرا می‌شود درحالی‌که در مرجع در ثانیه‌ی پنجم ربات به هدف همگرا می‌شود که این تفاوت به دلیل انتخاب ضرایب کنترلی در این شبیه‌سازی می‌باشد که باعث افزایش سرعت ربات و در نتیجه کوتاه‌تر شدن زمان رسیدن به هدف شده است. در نتیجه انتخاب ضرایب بسیار اهمیت دارد، با افزایش ضرایب v_0 و α زمان رسیدن به هدف کوتاه‌تر می‌شود و نمودار دارای جهش ناگهانی می‌شود که مناسب نمی‌باشد، به علاوه اگر v_0 را افزایش و α را کاهش دهیم ربات سریع‌تر به هدف نزدیک می‌شود اما به دلیل کاهش ضریب α دیرتر به هدف می‌رسد که مناسب نمی‌باشد زیرا زمان رسیدن به هدف طولانی می‌شود در حالی‌که سرعت در ابتدا زیاد است. در نهایت سرعت چرخ‌ها به صورت شکل ۱۵ بدست آمده است که تنها مقادیر آن با مرجع متفاوت است که دلیل آن تفاوت در انتخاب بهره‌ی کنترلی می‌باشد، همچنین در مقاله‌ی مرجع از روش کنترلی MPC استفاده شده است که در این روش می‌توان برای سرعت محدوده تعیین کرد که بازه‌ی آن بین -2 تا 2 می‌باشد اما در این مقاله انتخاب ضرایب به گونه‌ای بوده است که سرعت بین -8 تا 8 می‌باشد و به همین دلیل ربات در زمان کوتاه‌تری نسبت به مرجع به هدف رسیده است و به تبع آن سرعت همه‌ی چرخ‌ها در زمان کمتری به صفر همگرا شده‌اند.



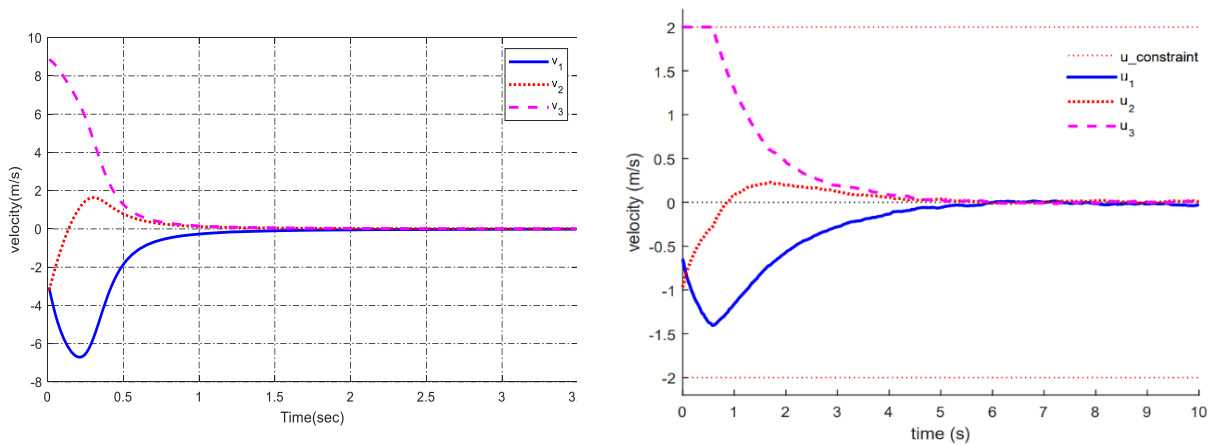
شکل ۱۲- حرکت ربات به سمت نقطه‌ی هدف



شکل ۱۳- دنبال کردن موقعیت ربات در مختصات دکارتی (شکل سمت راست مربوط به مرجع [۱۴] می‌باشد).



شکل ۱۴- دنبال کردن زاویه‌ی سر ربات (شکل سمت راست مربوط به مرجع [۱۴] می‌باشد).

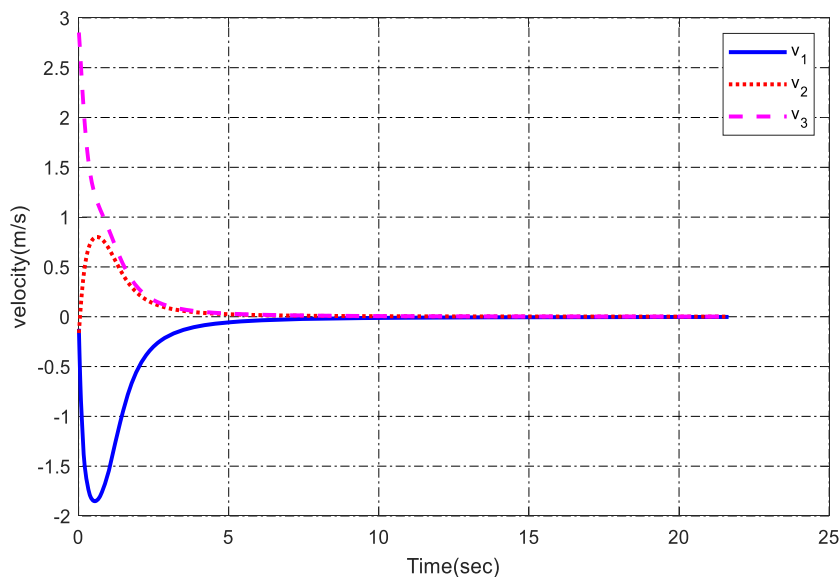


شکل ۱۵- سرعت‌های خطی ورودی سه چرخ (شکل سمت راست مربوط به مرجع [۱۴] می‌باشد).

در ادامه برای اینکه شبهه سازی انجام شده به محدوده‌ی سرعت ذکر شده در مرجع نزدیک شود پارامترها به صورت زیر در نظر گرفته می‌شود:

$$v_0 = 2, \alpha = 0.5$$

که در این صورت مدت زمان رسیدن به هدف افزایش می‌یابد و در این حالت سرعت چرخ اول در مدت ۰/۵ ثانیه از ۰/۲- به مقدار ۱/۸- کاهش می‌یابد در حالی که در مرجع سرعت چرخ اول از ۰/۷- به ۱/۵- کاهش می‌یابد در نتیجه شیب کاهش سرعت چرخ اول در ابتدا نسبت به مرجع بیشتر است و سپس در ثانیه‌ی ۱/۶ مانند مرجع به صفر می‌رسد، به علاوه سرعت چرخ دوم در این شبهه سازی در مدت ۰/۵ ثانیه از ۰/۲- به ۰/۸- افزایش می‌یابد در حالی که در مرجع در مدت ۲ ثانیه از مقدار ۱- به ۰/۲- افزایش می‌یابد به همین دلیل در ابتدا شیب افزایش سرعت چرخ دوم در این شبهه سازی بیشتر از مرجع می‌باشد. برای چرخ سوم مانند مرجع مدت ۵ ثانیه طول کشیده است تا سرعت به صفر کاهش یابد. همچنین به دلیل اینکه در مرجع نوع کنترل کننده از افزایش ناچیز سرعت از بازه‌ی تعیین شده می‌تواند جلوگیری کند، افزایش ابتدایی سرعت از محدوده‌ی ۲ را کنترل کرده است، در حالی که همان طور که در شکل (۱۶) مشاهده می‌شود سرعت چرخ سوم ابتدا از ۲/۸ شروع شده است. این تفاوت‌ها می‌تواند به دلیل نوع کنترل کننده و همچنین تفاوت در مقادیر پارامترهای ربات با مرجع باشد.



شکل ۱۶- سرعت‌های خطی ورودی سه چرخ

۶- شبیه سازی

۶-۱- عبور از موانع و رسیدن به هدف با حرکت انتقالی بدون دوران

در این مرحله با استفاده از کنترل رفتار مبنا ربات از موقعیت $(0,0)$ شروع به حرکت می‌کند و پس از عبور از ۴ مانع در موقعیت $(45,40)$ متوقف می‌شود و در طول حرکت زاویه‌ی سر ربات 90° درجه نسبت به افق باقی می‌ماند (شکل ۱۷). نتایج بدست آمده از سرعت خطی و زاویه‌ای ربات، گشتاور چرخ‌ها و مقایسه‌ی خطاهای کنترل کننده‌ی PID و فیدبک حالت در بخش (۶-۳) آورده شده است. پارامترهای موجود در روابط (۲۷-۲۰) برای شبیه سازی به صورت زیر در نظر گرفته شده است:

$$v_0 = 3, \alpha = 1, c = 70, \varepsilon = 0.2, \beta = 0.5$$

ضرایب کنترل کننده‌ی PID به صورت زیر در نظر گرفته شده است:

$$K_{\dot{x}P} = 300, K_{\dot{x}I} = 100$$

$$K_{\dot{y}P} = 200, K_{\dot{y}I} = 150$$

$$K_{\phi P} = 5, K_{\phi D} = 60$$

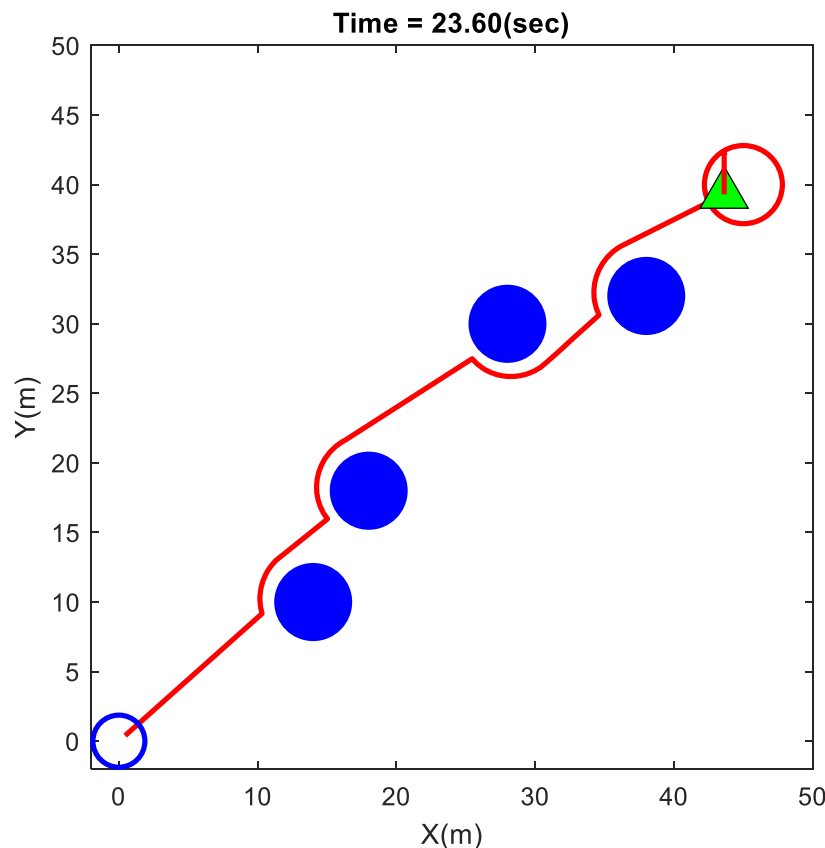
پس از بررسی کنترل پذیری سیستم با استفاده از دستور place، جایابی قطب‌های مطلوب انجام گرفته است. ماتریس ضریب (K) کنترل کننده‌ی فیدبک حالت به صورت زیر در نظر گرفته شده است:

$$K = \begin{bmatrix} -18769 & -11062 & 14207 & -434 & -253 & 341 \\ 8398 & -11313 & 14807 & 189 & -256 & 348 \\ 10850 & 22046 & 2034 & 251 & 506 & 47 \end{bmatrix}$$

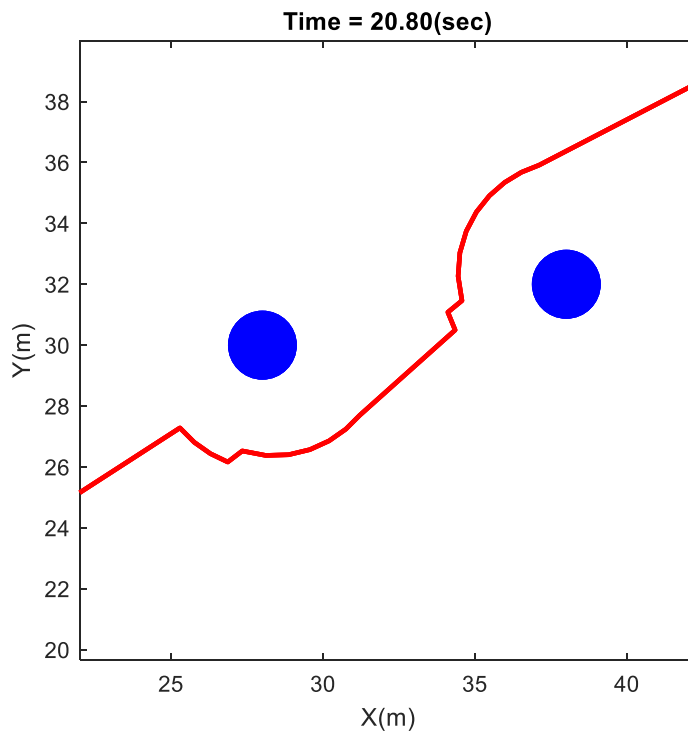
انتخاب پارامترها جهت تعیین ضرایب کنترلی (K_{GTG}, K_{AO}) مناسب بسیار اهمیت دارد، اگر مقدار ضریب v_0 از مقدار تعیین شده بسیار بزرگتر انتخاب شود، سرعت رفتار رفتن به هدف افزایش می یابد در حالی که سرعت عبور از موانع کمتر است که این رفتار مناسب نمی باشد زیرا باعث حرکات ناهموار و ناپیوسته می شود و بالعکس. به علاوه افزایش ۱۰۰ برابری ضریب α تاثیری در حرکت ربات ندارد و چنانچه این ضریب ۰/۰۱ برابر شود حرکت ربات هنگام عبور از مانع نزدیک هدف ناهموار می شود و به علاوه سرعت ربات هنگام نزدیک شدن به هدف به شدت کاهش می یابد و مدت زمان رسیدن به هدف بسیار افزایش می یابد که نامناسب می باشد.

اگر مقدار ضریب ε برای تعیین رفتار دوری از موانع بسیار بزرگتر از مقدار تعیین شده باشد، سرعت عبور از موانع کم می شود و به علاوه ربات هنگام عبور از مانعی که نزدیک هدف است به دام می افتد و حرکت ناهموار ایجاد می کند، اما کاهش این ضریب تاثیر محسوسی در روند حرکتی ربات ندارد و تنها مقدار کمی ناهمواری در هنگام عبور از موانع نزدیک هدف ایجاد می کند. اگر ضریب c را برای مثال از مقدار ۷۰ به ۱۰۰ افزایش دهیم باعث ایجاد حرکات ناهموار هنگام عبور از موانع می شود (شکل ۱۸). به علاوه اگر این ضریب ۰/۱ برابر شود سرعت عبور از موانع بسیار کاهش می یابد که مطلوب نمی باشد.

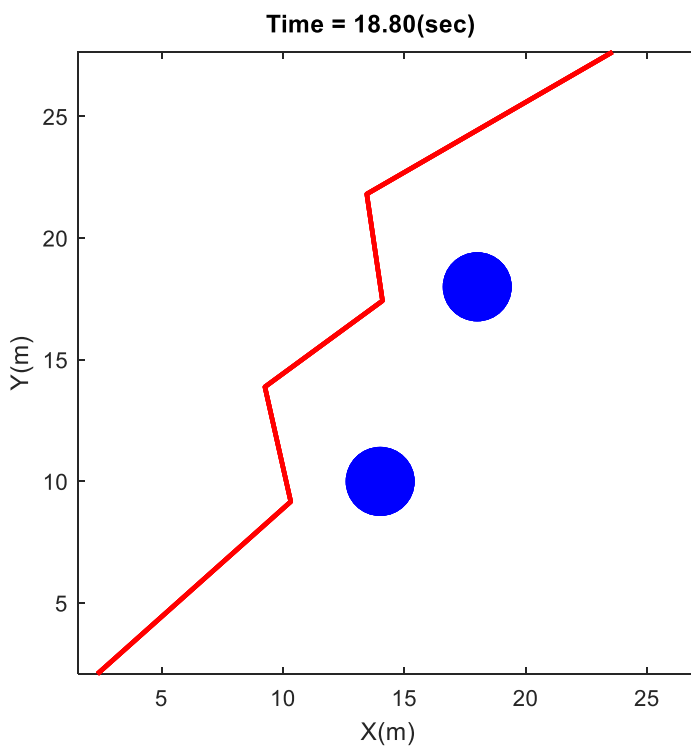
همچنین تعیین ضریب β برای رفتار دنبال کردن دیوار از اهمیت زیادی برخوردار است، برای مثال اگر این ضریب ۱۰ برابر مقدار تعیین شده، شود هنگام عبور از موانع به صورت رفتار دنبال کردن دیوار، حرکت به نرمی انجام نمی گیرد که نامناسب می باشد (شکل ۱۹) و چنانچه این ضریب ۰/۱ برابر شود سرعت عبور از موانع بسیار کاهش می یابد. در نهایت با در نظر گرفتن موارد فوق بهترین انتخاب انجام گرفته است.



شکل ۱۷- شبه سازی حرکت ربات با جهت گیری ثابت



شکل ۱۸- عبور از موانع با $c = 100$



شکل ۱۹- عبور از موانع با $\beta = 5$

۶-۲- عبور از موانع و رسیدن به هدف با حرکت انتقالی و دورانی به طور همزمان و مستقل از هم

۶-۲-۱- کنترل دو ربات با هدف جهت گیری به سمت یکدیگر

با استفاده از الگوریتم ذکر شده رفتار ربات‌ها به گونه‌ای کنترل می‌شوند که همواره سر دو ربات در طول مسیر حرکت به سمت یکدیگر باشد و همچنین علاوه بر تنظیم ضرایب برای سرعت های خطی ربات، ضریب K_{PW} را برای کنترل سرعت زاویه‌ای ربات می‌توان تنظیم کرد که با افزایش آن سر ربات سریع‌تر به سمت ربات دیگر می‌چرخد. همچنین در این شبیه سازی مانع دیوار اضافه شده است که برای دنبال کردن آن نیاز به معادله‌ی خطوطی داریم که اضلاع دیوار را تشکیل می‌دهند (شکل ۲۰). نتایج مربوط به سرعت های خطی و زاویه‌ای دو ربات و گشتاور چرخ‌ها در بخش (۳-۶) آورده شده است.

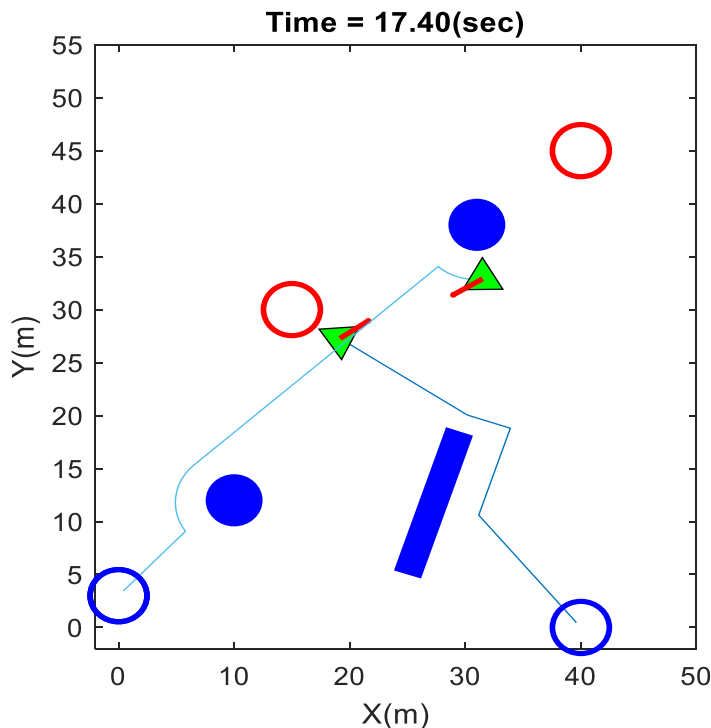
$$\begin{aligned} \phi_{d1} &= \text{atan2}((y_2 - y_{01}), (x_2 - x_{01})) \\ \phi_{d2} &= \text{atan2}((y_1 - y_{02}), (x_1 - x_{02})) \\ w_1 &= K_{PW1}(\text{atan2}(\sin(\phi_{d1} - \phi_1), \cos(\phi_{d1} - \phi_1))) \\ w_2 &= K_{PW2}(\text{atan2}(\sin(\phi_{d2} - \phi_2), \cos(\phi_{d2} - \phi_2))) \end{aligned}$$

پارامترهای موجود در روابط (۲۷-۲۰) برای شبیه سازی ربات اول به صورت زیر در نظر گرفته شده است:

$$v_{01} = 3, \alpha_1 = 4, c_1 = 45, \varepsilon_1 = 0.3, \beta_1 = 1, K_{PW1} = 2$$

پارامترهای موجود در روابط (۲۷-۲۰) برای شبیه سازی ربات دوم به صورت زیر در نظر گرفته شده است:

$$\begin{aligned} v_{02} &= 3, \alpha_2 = 4, c_2 = [40 \ 1] \\ \varepsilon_2 &= [0.5 \ 0.2], \beta_2 = 1, K_{PW2} = 2 \end{aligned}$$



شکل ۲۰- شبیه سازی حرکت دو ربات

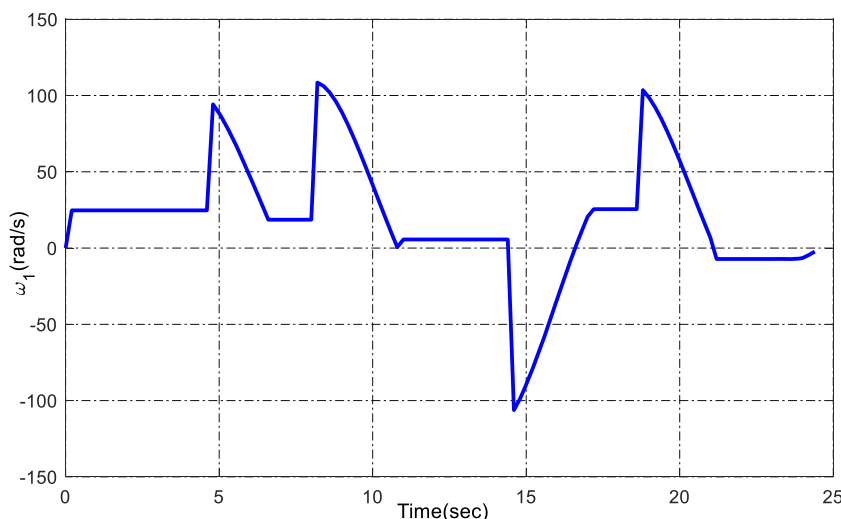
۳-۶- نتایج شبیه سازی

۱-۳-۶- نتایج شبیه سازی اول

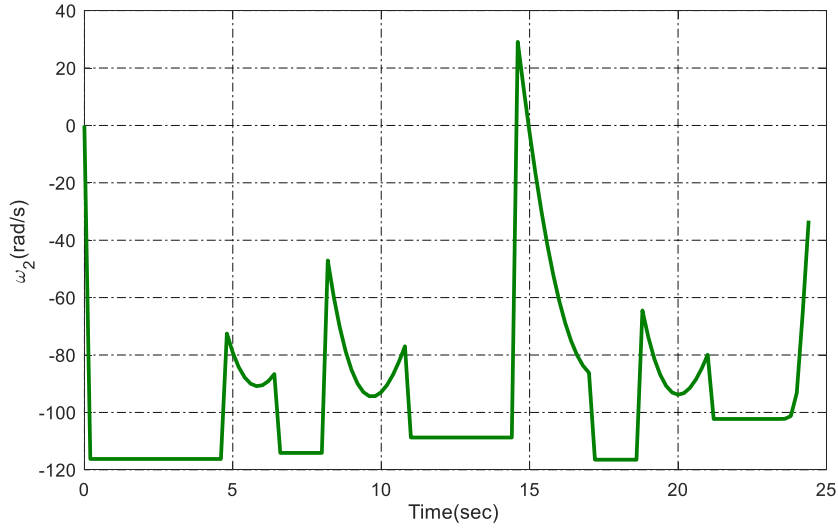
در شبیه سازی اول کنترل ربات جهت رسیدن به هدف و دوری از موانع با استفاده از کنترل رفتار مبنا انجام شده است و سپس نتایج دو کنترل کننده PID و فیدبک حالت در دقت دنباله روی با هم مقایسه شده است. پس از شبیه سازی با استفاده از روابط بخش (۲) سرعت‌های زاویه‌ای چرخ‌ها (شکل‌های ۲۳-۲۱) و سپس گشتاور آن‌ها (شکل‌های ۲۴-۲۶) بدست آمده است.

همان طور که در شکل‌های زیر مشخص است، هنگامی که ربات به سمت هدف می‌رود، گشتاور ثابتی دارد و به دلیل سوئیچ بین رفتار رفتن به هدف و دنبال کردن دیوار، گشتاور به طور ناگهانی تغییر می‌کند. اما در هنگام دنبال کردن دیوار به نرمی رفتار می‌کند و حرکت هموار می‌باشد و هر حرکت به طور کامل نیز انجام می‌گیرد. به علاوه مشاهده می‌شود که نمودارهای گشتاور و سرعت زاویه‌ای هر چرخ شبیه هستند زیرا با توجه به روابط موجود در بخش (۲)، گشتاور به سرعت زاویه‌ای چرخ، شتاب زاویه‌ای چرخ و نیروی محرکه‌ی آن بستگی دارد و به دلیل آنکه شتاب زاویه‌ای به جز در شروع تغییر جهت ربات صفر است و نیروی محرکه نیز تنها در چرخش‌های ناگهانی کمی تغییر می‌کند و در موارد دیگر نزدیک صفر است، این دو نمودار شبیه به هم هستند و تنها مقادیر آن‌ها با هم متفاوت است زیرا سرعت زاویه‌ای در ضریب C ضرب و به ضریب k نیز تقسیم می‌شود که این دو ضریب در جدول (۲) تعریف شده‌اند.

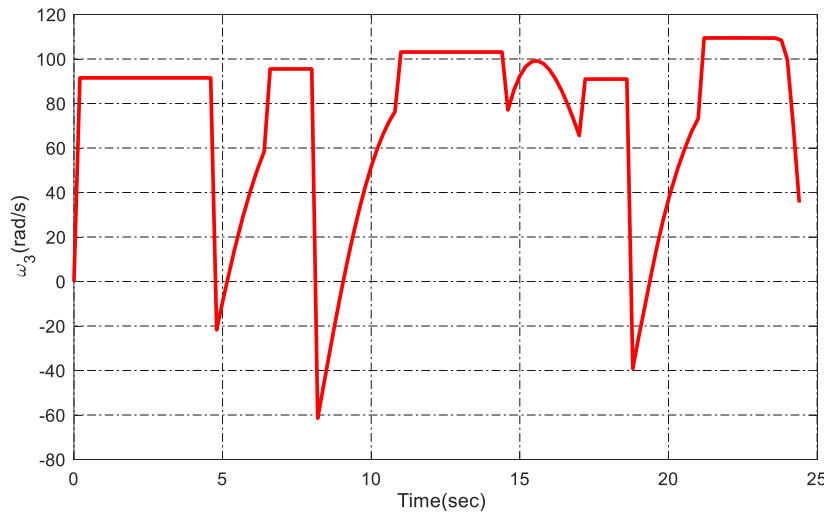
همان طور که در شکل‌های (۲۷-۲۹) مشاهده می‌شود، عملکرد کنترل کننده PID نسبت به فیدبک حالت بسیار مناسب‌تر است که نشان دهنده‌ی آن است که استفاده از گشتاور چرخ‌ها جهت کنترل سرعت و جهت گیری ربات موجب دقت دنباله روی می‌شود و عملکرد کنترل کننده فیدبک حالت در عبور از موانع بسیار نامناسب و ناپایدار است. برای مثال خطای سرعت در جهت x در $t = 5sec$ در کنترل کننده PID حدود 0.33 است که بسیار ناچیز است در حالی که خطا در کنترل فیدبک حالت حدود 0.3 می‌باشد و همچنین در کنترل کننده PID خطای زاویه‌ی سر ربات صفر است که مطلوب می‌باشد و به طور کلی در کنترل فیدبک حالت در هنگام روبرویی با موانع رفتار ربات ناپایدار می‌باشد.



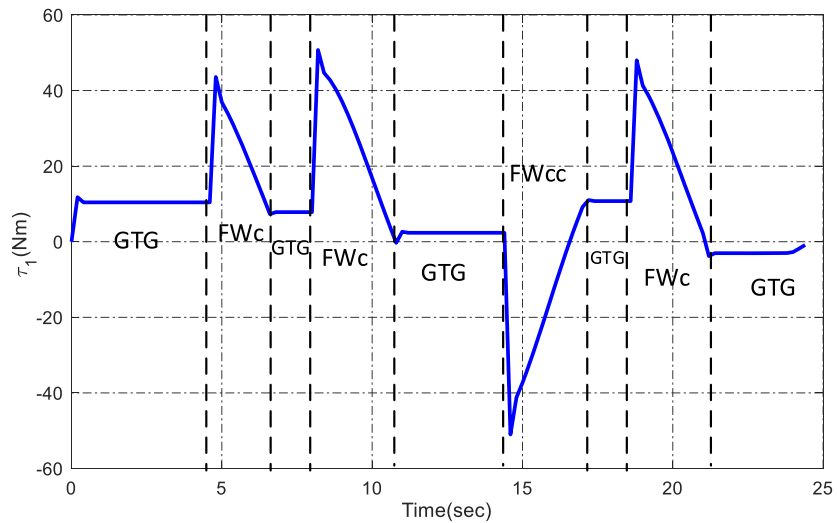
شکل ۲۱- سرعت زاویه‌ای چرخ اول



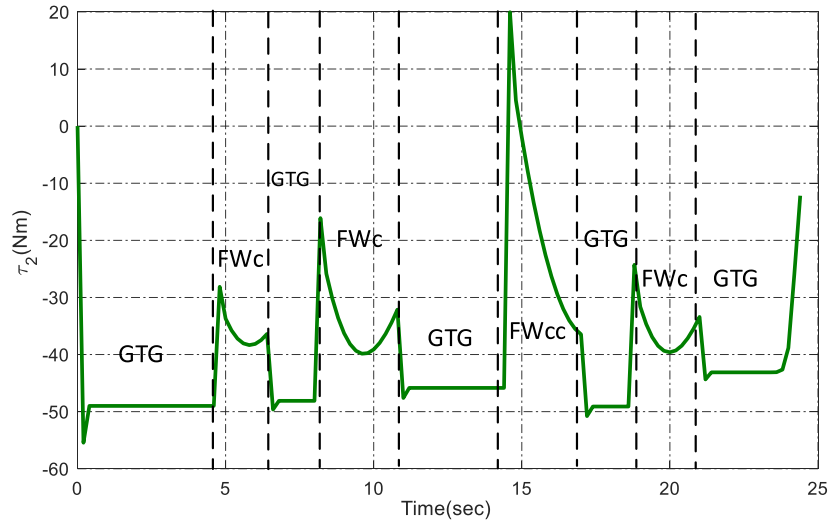
شکل ۲۲- سرعت زاویه‌ای چرخ دوم



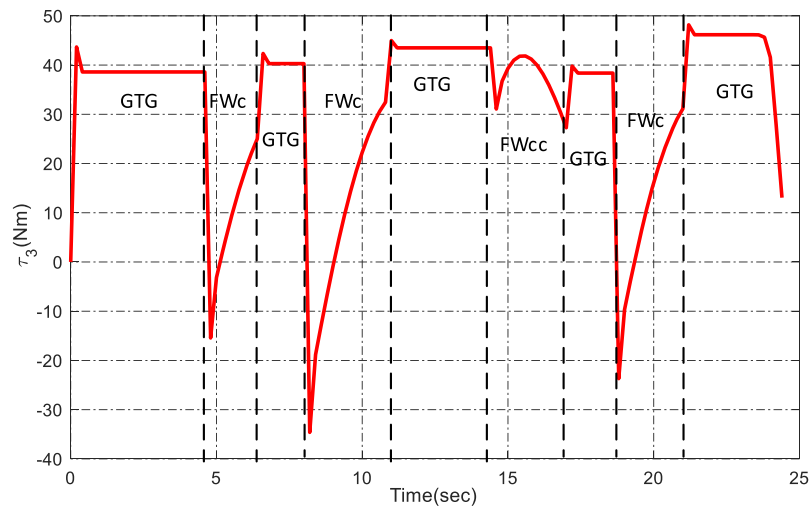
شکل ۲۳- سرعت زاویه‌ای چرخ سوم



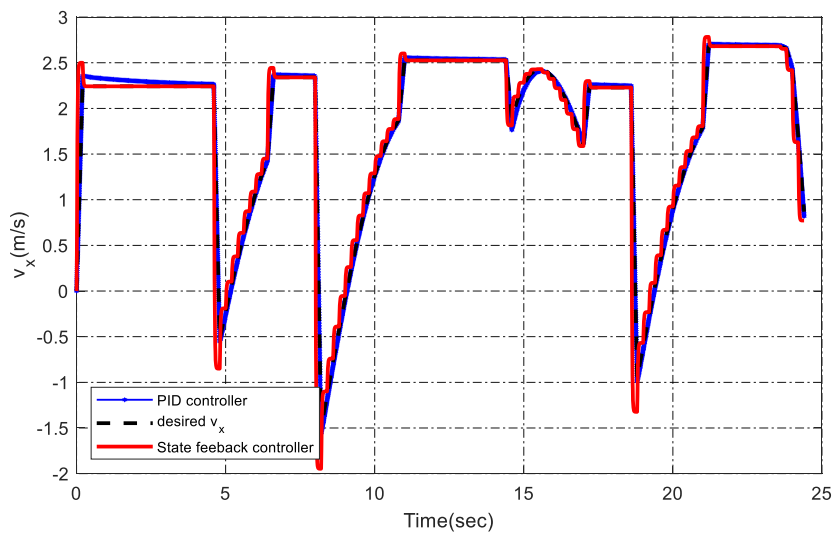
شکل ۲۴- گشتاور چرخ اول



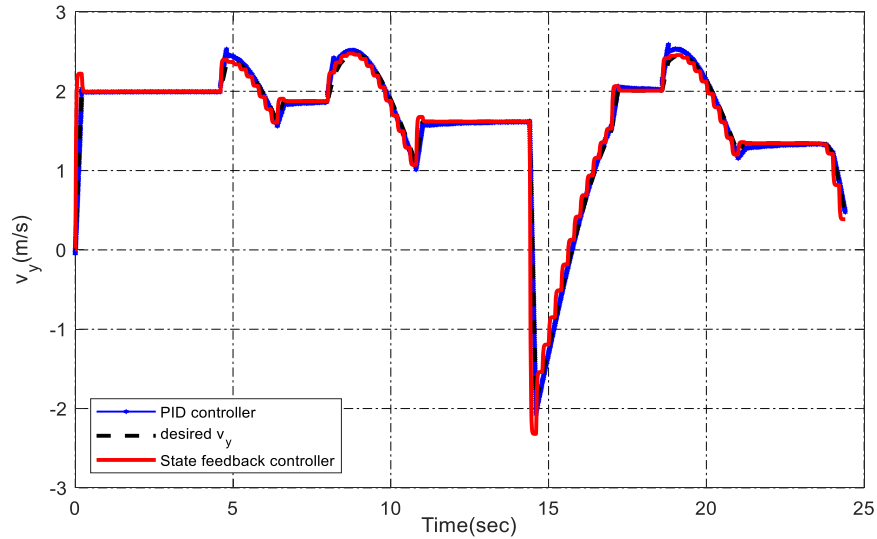
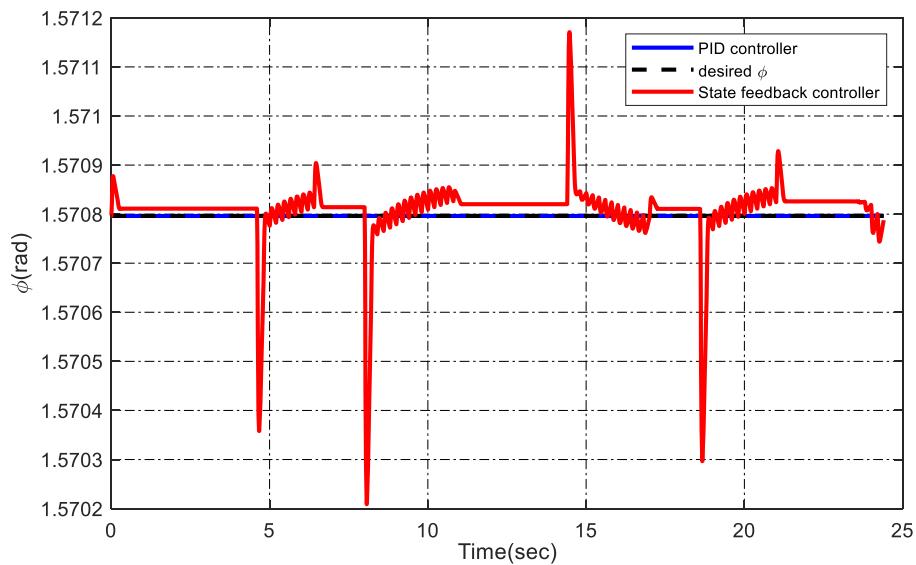
شکل ۲۵- گشتاور چرخ دوم



شکل ۲۶- گشتاور چرخ سوم



شکل ۲۷- سرعت ربات در جهت X

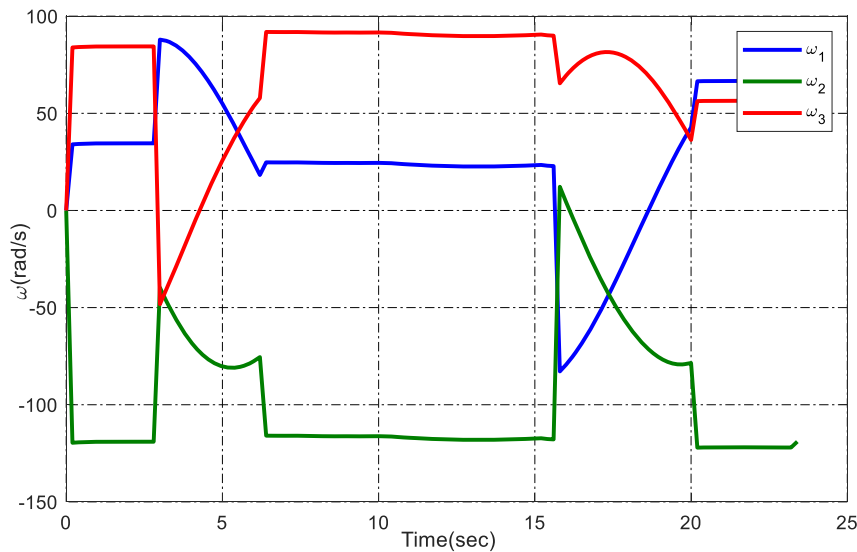
شکل ۲۸- سرعت ربات در جهت y 

شکل ۲۹- زاویه‌ی سر ربات

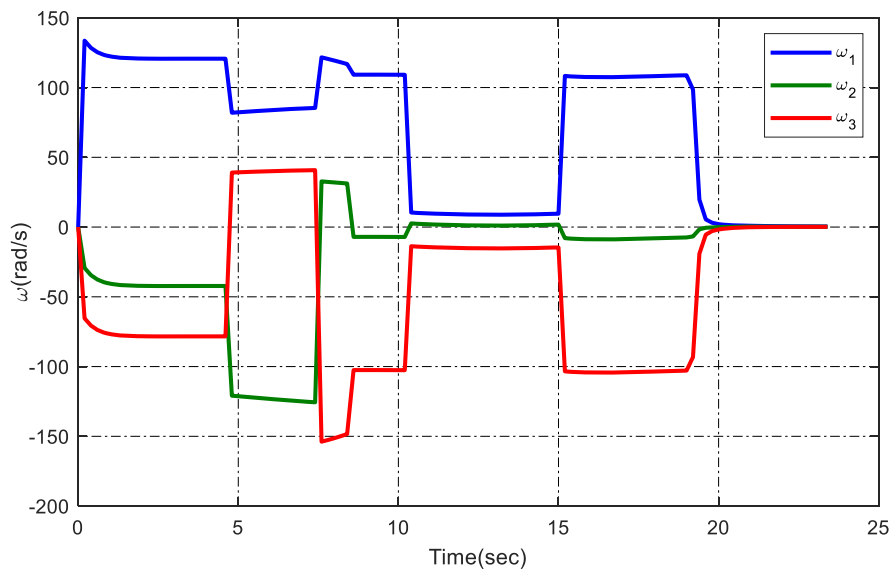
۲-۳-۶- نتایج شبه سازی دوم

ابتدا پس از کنترل حرکت ربات با استفاده از کنترل رفتار مبنای، از روابط سینماتیکی سرعت‌های زاویه‌ای و گشتاور چرخ‌ها (شکل‌های ۳۰-۳۳) بدست آمده است. رفتار ربات اول شبه به بخش قبل می‌باشد زیرا موانع دایره‌ای شکل هستند و تنها سر ربات به جای جهت‌گیری به سمت مانع که یک نقطه‌ی ثابت است، یک نقطه‌ی متحرک یعنی ربات دوم را دنبال می‌کند. تحلیل نمودارها مانند حالت‌های قبل می‌باشد، در تحلیل نمودار ربات دوم، نکات مهمی در رابطه با عبور ربات از مانع دیوار می‌باشد، همان‌طور که در شکل مشخص است برای مثال در چرخ اول ابتدا گشتاور حدود ۵۰ نیوتن متر است و ربات در ثانیه‌ی ۵ام به دیوار نزدیک می‌شود و وارد رفتار دنبال کردن دیوار می‌شود و کاهش ناگهانی در گشتاور رخ می‌دهد و در هنگام عبور از

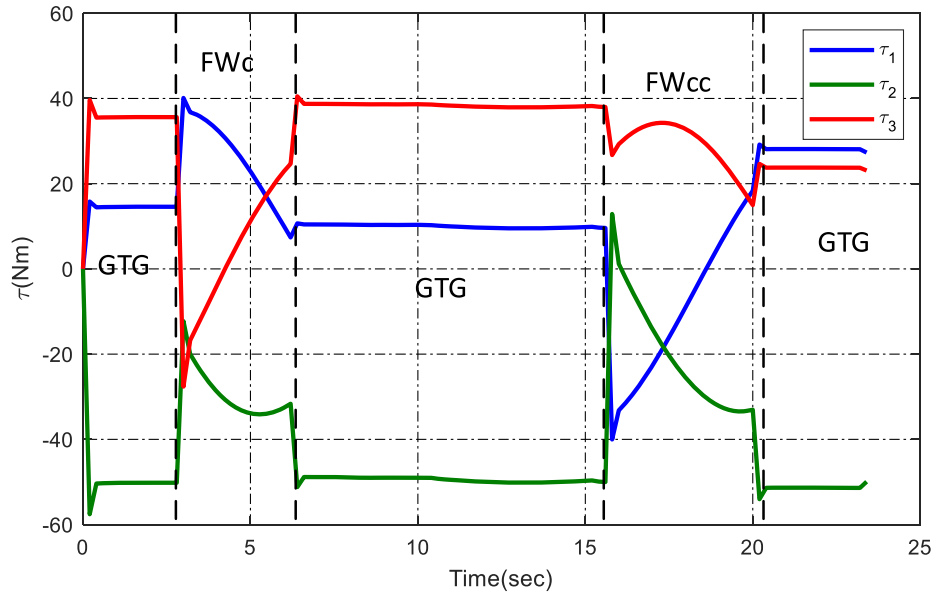
دیوار گشتاور آن به مقدار ثابت ۳۵ نیوتن متر می‌رسد و از ضلع اول در ثانیه‌ی ۷ام عبور می‌کند و دوباره برای عبور از ضلع دوم دیوار گشتاور آن افزایش می‌یابد و به مقدار ۵۰ نیوتن متر می‌رسد و پس از عبور کامل از دیوار در ثانیه ۹ تا ۱۱ مجدداً وارد رفتار رفتن به هدف می‌شود و گشتار آن به مقدار ثابت ۴۵ می‌رسد و در ثانیه‌ی ۱۱ام ربات دوم برای جلوگیری از برخورد با ربات اول متوقف می‌شود و ربات از ثانیه‌ی ۱۰ تا ۱۵ تنها دوران دارد و موقعیت x و y آن تغییر نمی‌کند و بنابراین ربات در این مدت سرعت خطی ندارد و تنها سرعت زاویه‌ای دارد و به همین دلیل سرعت زاویه‌ای چرخ در این مدت کاهش می‌یابد و به مقدار ۸/۸ می‌رسد و گشتاور به ۳/۷ کاهش می‌یابد، پس از دور شدن از ربات اول دوباره به رفتار رفتن به هدف برمی‌گردد و میزان گشتاور چرخ اول دوباره افزایش می‌یابد و به مقدار ۴۵ نیوتن متر می‌رسد و هنگامی که به هدف می‌رسد متوقف می‌شود و تنها سرعت زاویه‌ای بسیار ناچیزی دارد، به همین دلیل گشتاور همه‌ی چرخ‌ها تقریباً به صفر می‌رسد.



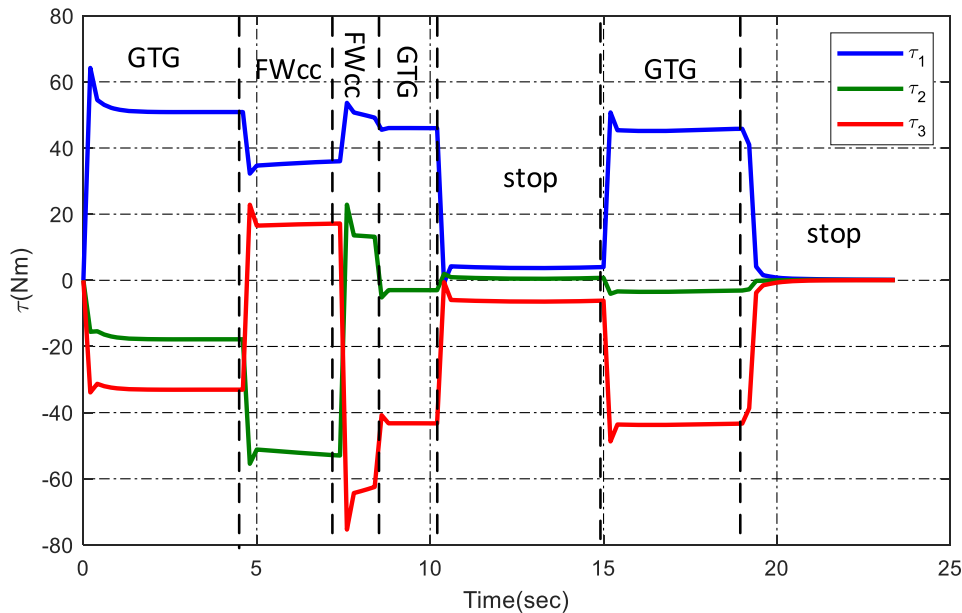
شکل ۳۰- سرعت زاویه‌ای چرخ‌های ربات اول



شکل ۳۱- سرعت زاویه‌ای چرخ‌های ربات دوم



شکل ۳۲ - گشتاور چرخ های ربات اول



شکل ۳۳ - گشتاور چرخ های ربات دوم

همان طور که در شکل های بالا مشاهده می شود، گشتاور ربات در هنگام عبور از مانع دایره ای به آرامی کاهش یا افزایش می یابد در حالی که در عبور از مانع دیوار گشتاور مقدار ثابتی باقی می ماند زیرا شیب مسیر عبوری از دیوار ثابت است اما در دایره متغیر است و در نهایت تاثیر آن در گشتاور مشاهده می شود. لازم به ذکر است که به دلیل سوئیچ بین رفتار رفتن به هدف و رفتار دنبال کردن دیوار، گشتاور به طور ناگهانی تغییر می کند. در این مقاله پس از استخراج روابط سینماتیکی و دینامیکی ربات، از روش کنترل رفتار مبنا به منظور جهت یابی حرکت یک ربات متحرک همه جهته ی سه چرخ استفاده شده است و چندین شبیه سازی از جمله جهت گیری دو ربات به سمت یکدیگر که همراه با عبور از موانع و تغییر وضعیت بین هر رفتار و رسیدن به

نقطه‌ی هدف می‌باشد انجام گرفته است. به علاوه از گشتاور چرخ‌ها به عنوان ورودی به ربات با استفاده از کنترل کننده‌ی PID جهت کنترل حرکت مطلوب ربات و دنباله روی سرعت و جهت‌گیری مطلوب آن استفاده شده است و نتایج بدست آمده در مقایسه با کنترل کننده‌ی فیدبک حالت، نشان دهنده‌ی عملکرد بهتر کنترل کننده‌ی PID می‌باشد و در ادامه این روش کنترلی با سایر مراجع صحت سنجی شده است و مشاهده شده است که کنترل کننده‌ی PID به کار برده شده با خطای کمتری نسبت به مرجع [۱۶] عمل کرده است. همچنین در ادامه‌ی صحت سنجی روش کنترل رفتار مبنا با شبیه سازی مرجع [۱۴] مقایسه شده است که نتایج حاصل نشان دهنده‌ی عملکرد مطلوب این روش برای جهت یابی ربات می‌باشد. به طور کلی مهم‌ترین مزیت این روش نسبت به روش‌هایی مانند میدان پتانسیل گیر نکردن ربات در حداقل موضعی می‌باشد و با وجود موانع با اشکال پیچیده و نزدیک به هم این روش به درستی عمل می‌کند.

مراجع

- [1] Tanner, H.G., Loizou, S.G., and Kyriakopoulos, K.J., "Nonholonomic Navigation and Control of Cooperating Mobile Manipulators", IEEE Transactions on Robotics and Automation, Vol. 19(11), pp. 53-64, (2003).
- [2] He, W., Chen, Y., and Yin, Z., "Adaptive Neural Network Control of an Uncertain Robot with Full-state Constraints", IEEE Transactions on Cybernetics", Vol. 46(3), pp. 620-629, (2015).
- [3] He, W., Dong, Y., and Sun, C., "Adaptive Neural Impedance Control of a Robotic Manipulator with Input Saturation", IEEE Transactions on Systems, Man, and Cybernetics: Systems, Vol. 46(3), pp. 334-344, (2016).
- [4] Cui, R., Zhang, X., and Cui, D., "Adaptive Sliding-mode Attitude Control for Autonomous Underwater Vehicles with Input Nonlinearities", Ocean Engineering. Vol. 123, pp. 45-54, (2016).
- [۵] فومنی، س.، "تحلیل سینماتیکی-سینتیکی پیمایش مستقیم الخط ربات انسان نما"، نشریه مدل سازی در مهندسی، دوره ۳، شماره ۱۷، (۱۳۸۸).
- [6] Cui, R., Guo, J., and Mao, Z., "Adaptive Backstepping Control of Wheeled Inverted Pendulums Models", Nonlinear Dynamics, Vol. 79(1), pp. 501-511, (2015).
- [۷] موسویان، ع.ا.، حسینی، ش.، "طراحی پایدارترین حرکت ربات متحرک در مسیر مشخص"، نشریه مدل سازی در مهندسی، دوره ۱۱، شماره ۳۳، صفحه ۱۴-۱، (۱۳۹۲).
- [۸] مرادی، م.، نیکوبین، ا.، آزادی، س.، "طراحی مسیر بهینه و بالانسینگ تکراری برای ربات متحرک در حرکات سریع"، نشریه مدل سازی در مهندسی، دوره ۱۴ شماره ۴۷، صفحه ۱۵۳-۱۴۱، (۱۳۹۵).
- [9] Moreno, J., Clotet, E., Lupiañez, R., Tresanchez, M., Martinez, D., Pallejà, T., Casanovas, J., and Palacín, J., "Design, Implementation and Validation of the Three-wheel Holonomic

- Motion System of the Assistant Personal Robot (APR)", *Sensors*, Vol. 16(10), pp. 1658, (2016).
- [10] Alshorman, A.M., Alshorman, O., Irfan, M., Glowacz, A., Muhammad, and F., Caesarendra, W., "Fuzzy-based Fault-tolerant Control for Omnidirectional Mobile Robot", *Machines*, Vol. 8(3), pp. 55, (2020).
- [11] Hacene, N., and Mendil, B., "Fuzzy Behavior-based Control of Three Wheeled Omnidirectional Mobile Robot", *International Journal of Automation and Computing*, Vol. 16(2), pp. 163-185, (2019).
- [12] Liu, X., Chen, H., Wang, C., Hu, F., and Yang, X., "MPC Control and Path Planning of Omni-directional Mobile Robot with Potential Field Method", In *International Conference on Intelligent Robotics and Applications, Lecture Notes in Computer Science*, Vol. 10985, Springer, Cham, pp.170-181, (2018).
- [13] Armah, S., Yi, S., and Abu-Lebdeh, T., "Implementation of Autonomous Navigation Algorithms on Two-wheeled Ground Mobile Robot", *American Journal of Engineering and Applied Sciences*, Vol. 7(1), pp. 149-164, (2014).
- [14] Wang, C., Liu, X., Yang, X., Hu, F., Jiang, A., and Yang, C., "Trajectory Tracking of an Omni-directional Wheeled Mobile Robot using a Model Predictive Control Strategy", *Applied Sciences*, Vol. 8(2), pp. 231, (2018).
- [15] Watanabe, K., "Knowledge-based Intelligent Electronic Systems. Proceedings KES'98, Control of an Omnidirectional Mobile Robot, in 1998 Second International Conference, IEEE, April 21-23, Adelaide, SA, Australia, (1998).
- [16] Watanabe, K., Shiraishi, Y., Tzafestas, S.G., Tang, J., and Fukuda, T., "Feedback Control of an Omnidirectional Autonomous Platform for Mobile Service Robots", *Journal of Intelligent and Robotic Systems*, Vol. 22(3), pp. 315-330, (1998).
- [17] Tang, L., Dian, S., Gu, G., Zhou, K., Wang, S., and Feng, X., "A Novel Potential Field Method for Obstacle Avoidance and Path Planning of Mobile Robot", *3rd International Conference on Computer Science and Information Technology, IEEE*, July 9-11, Chengdu, China (2010).
- [18] Wang, W., Zhu, M., Wang, X., He, S., He, J., and Xu, Z., "An Improved Artificial Potential Field Method of Trajectory Planning and Obstacle Avoidance for Redundant Manipulators", *International Journal of Advanced Robotic Systems*, Vol. 15(5), pp. 172988141879962, (2018).
- [19] Li, D., Pan, Z., and Deng, H., "Two-dimensional Obstacle Avoidance Control Algorithm for Snake-like Robot in Water Based on Immersed Boundary-lattice Boltzmann Method and Improved Artificial Potential Field Method", *Transactions of the Institute of Measurement and Control*, Vol. 42(10), pp. 1840-1857, (2020).
- [20] Iwatsuki, M., Nakano, K., and Ohuchi, T., "Target Point Tracking Control of Robot Vehicle by Fuzzy Reasoning", *Transactions of the Society of Instrument and Control Engineers*, Vol. 27(1), pp. 70-76, (1991).

- [21] Saito, M., and Tsumura, T., "Collision Avoidance Among Multiple Mobile Robots", Transactions of the Institute of Systems, Control and Information Engineers, Vol. 3(8), pp. 252-260, (1990).

Abstract

The main purpose of this article is tracking the trajectory of the omni-directional robot using the navigation algorithm by behavior-based control and a new simulation with the aim of orienting the two robots toward each other. Using the robot kinematics to obtain the equations related to the speed and torque of the wheels. in this paper, the robot platform structure has three sets of orthogonal omni wheels that can move in all directions with the ability of displacement without rotation or rotation and displacement simultaneously. the appropriate dynamic equation of the robot has been extracted, then using inverse dynamics of the robot to control the robot's path. In addition, validate some simulation by presented method with other references. At the end, the results obtained from the simulations are discussed and analyzed.