گزارش پروژه‌ی پایان‌نیم سال درس رباتیک

داریوش حسن‌پور آده

۹۳۰۸۱۶۴

۱ مقدمه

طبق تعریف اولیه پروژه مبنی بر تعمییر و راه‌اندازی ربات مین‌یاب آزمایشگاه هوش‌مصنوعی به علت محدودیت‌های سخت افزاری /معماری ربات مذکور و عدم امکان *مانورو برای پیاده‌سازی حداقل یکی از الگوریتم های درس رباتیک بروی ربات پروژه ی ثانویه جهت شبیه‌سازی رفتار ربات در صورتی که ربات توانایی سرعت دهی دلخواه به موتورهای خود را داشته باشد تخصیص داده شد. که در این گزارش به شرح هر یک از دو پروژه ی انجام شده می‌پردازیم.*

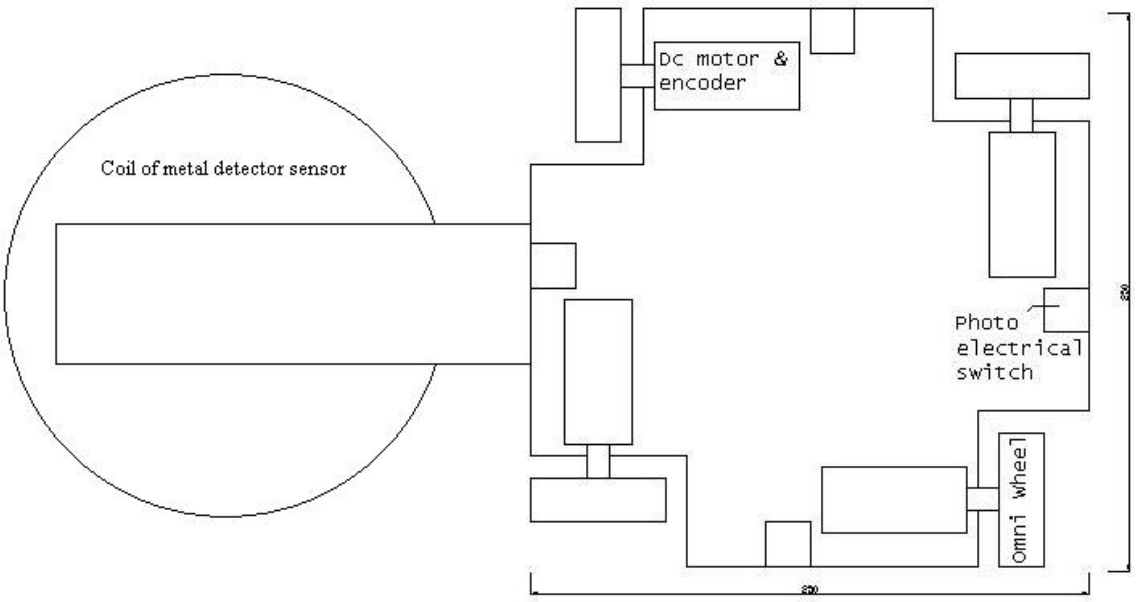
فهرست مطالب

۲ تعمییر و راه اندازی سخت‌افزاری ربات

۱.۲ سخت‌افزارهای ربات

این ربات که در شکل ۱ نمایش ساختاری ربات آمده است؛ دارای مشخصات سخت‌افزاری زیر بوده است:

- *۱ عدد شاسی*
- *۴ عدد موتور*
- *۴ عدد چرخ سوئدی (با $\gamma = \frac{\pi}{۴}$)*
- *۴ عدد سنسور مادون‌قرمز با حساسیت ۱۰ سانتی متر*
- *۱ عدد برد*
- *۲ عدد زیرپردازنده*
- *۱ عدد نمایشگر با آدرس‌دهی ۴ بیتی*
- *۱ عدد حسگر فلزیاب (معیوب)*



شکل۱: نمایش ساختاری ربات

۲.۲ شرح عملکرد ربات

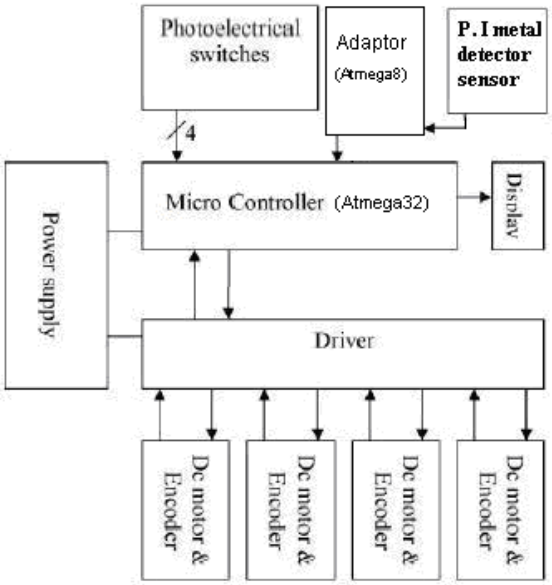
این ربات دارای ۴ چرخ دیفرانسیلی می‌باشد و هریک از چرخ‌های میتواند به صورت جدا دستورات سرعت متفاوتی را اجرا کنند ولی متاسفانه سازندگان این ربات مدارها را طوری سیم پیچی کرده اند که هر یک از چرخ

ها میتواند فقط ۱ سرعت بگیرند(یا حرکت می‌کنند یا حرکت نمی‌کنند!) بنابراین امکان ایجاد حرکات زاویه دار که نیازمند تخصیص سرعت های متفاوتی به چرخ‌ها می‌باشند را ندارد.

برای کنترل ربات یک ریزپردازنده Atmega32 استفاده شده است که تمامی سیگنال های حسگرها و دستورات کنترلی چرخ‌ها از طریق این ریزپردازنده کنترل میشوند. یک زیرپردازنده ی Atmega8 نیز برای انتزاع سازی علمیات مربوط به حسگر فلزیاب نیز تعبیه شده است که عملیات سطح پایین مربوط به راه‌اندازی، ارتباط و دریافت اطلاعات از حسگر فلزیاب را به عهده دارد و در انتها توسط یک وقفه به ریزپردانده اصلی(Atmega32) فرستاده میشود که تشخیص فلز را اعلام می‌کند.

۳.۲ شرح اتصالات اجزای الکترونیکی ربات

همانطور که میدانیم ریزپردازنده ی Atmega32 دارای ۴ عدد درگاه[۱] با نام های انتزاعی A, B, C, D می باشد که هر یک دارای ۸ عدد اتصال[۲] می‌باشند که در شکل ۲ شمای کلی اتصالات ربات آمده است و در این قسمت به معرفی اتصالات این درگاه‌ها با سایر قسمت‌های مدار می‌پردازیم.



شکل ۲: نمایش شماتیکی ربات

۱.۳.۲ اتصالات درگاه A

درگاه A اختصاصا برای اتصالات راه‌اندازه[۳]های موتورها استفاده شده است. که به شرح جدول ۱ می باشد.

---

[۱] Port
[۲] Pin
[۳] Driver

Pin	Connection	Target Motor
A_0	Input 1 Of Driver 1	Motor 2
A_1	Input 2 Of Driver 1	
A_2	Input 3 Of Driver 1	Motor 1
A_3	Input 4 Of Driver 1	
A_4	Input 1 Of Driver 2	Motor 3
A_5	Input 2 Of Driver 2	
A_6	Input 3 Of Driver 2	Motor 4
A_7	Input 4 Of Driver 2	

جدول۱: مشخصات اتصالات درگاه A

۲.۳.۲ اتصالات درگاه B

این درگاه برای کار با حسگرهای مادون‌قرمز مورد استفاده قرار گرفته اند. که شامل اتصالات ۰, ۱, ۲ و ۳ می‌باشند که به ۴ حسگر بعنوان ورودی حسگرها فرستاده شده است.

۳.۳.۲ اتصالات درگاه C

از اتصالات درگاه C فقط اتصالات ۰ و ۲ برای بازنشانی و فعال کردن نمایشگر استفاده شده است.

۴.۳.۲ اتصالات درگاه D

اتصالات درگاه D به صورت جدول ۲ است.

Pin	Connection	Description
D_0	NOT USED	N/A
D_1	NOT USED	
D_2	Sensor Input 0-2	Interrupt
D_3	Sensor Input 1-3	
D_4	LCD Output	Mapped accordingly to LCD's PortC4...7
D_5	LCD Output	
D_6	LCD Output	
D_7	LCD Output	

جدول۲: مشخصات اتصالات درگاه D

توجه شود که علت اینکه ۲ عدد وقفه برای ۴ عدد حسگر استفاده شده است این است که هر ۲ حسگر جلو و عقب به یک اتصال وقفه می‌دهند و بدین صورت که در حالت عادی فقط یکی از این حسگرها عمل میکنند و براحتی از سرعت چرخ‌ها میتوان حدس زد که کدام حسگر وقفه را ارسال کرده است و همین مساله برای حسگرهای چپ و راست نیز صادق است.

۵.۳.۲ برنامه‌ریزی ربات

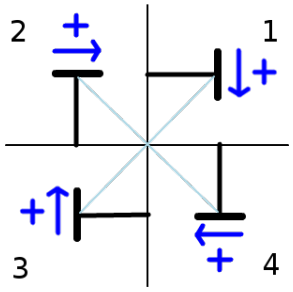
برای برنامه‌ریزی ربات و کنترل ربات فقط کافی است که ریزپردازنده *Atmega32* را برنامه ریزی نماییم. که برنامه ی نوشته شده به زبان *AVR* بوده و توسط کامپایلر *avr-gcc (GCC) 4.8.2* تحت سیستم عامل *Linux 3.13.0-34-generic* کامپایل گردیده است و توسط *avrdude 6.0.1* به داخل ریزپردازنده سوخته [۴] شده است. که توضیح راجع به روند سوزاندن و کدهای سوخته شده خارج از حوصله ی این نوشتار است.

۳ شبیه‌سازی نرم‌افزاری ربات

برای پیاده‌سازی شبیه‌سازی مورد نیاز برای هدایت روبات به مختصات دلخواه نیاز به نوشتن روابط سینماتیکی مورد نیاز داریم.[۱] برای نوشتن روابط سینماتیکی ربات ابتدا باید آنرا مدل‌سازی کنیم.

۱.۳ مدل‌سازی ریاضی ربات

برای مدل‌سازی اینطور در نظر گرفتیم که هر چرخ با چرخ دیگر دارای اختلاف درجه $\frac{\pi}{2}$ بوده است و با هر یک از محورهای محلی خود دارای زاویه $\frac{\pi}{4}$ هستند. که در شکل ۳ آمده است. همانطور که در شکل ۳ میبینید جهت چرخش **مثبت** چرخ‌ها آمده است و چرخ‌ها به ترتیب نواحی مثلثاتی که در آن قرار دارند شماره گذاری شده اند.



شکل۳: نمایش مدلی ربات برای شبیه‌سازی ربات در متلب

[۴] Burn

برای بدست‌آوردن محدودیت‌های سینماتیکی تحمیل شده از طرف چرخ‌ها نیاز داریم ویژگی های هندسی چرخ‌ها را معین کنیم که در جدول ۳ آمده است. در جدول ۳ مقادیر r و l را باتوجه به مقادیر داده شده به تمرین دوم این درس در نظر گرفته شده‌اند.

Wheel#	α	β	γ	l	r
1	$\frac{\pi}{4}$	$-\frac{\pi}{4}$	$\frac{\pi}{2}$	0.25	0.125
2	$\frac{3\pi}{4}$	$-\frac{\pi}{4}$	$\frac{\pi}{2}$	0.25	0.125
3	$\frac{5\pi}{4}$	$-\frac{\pi}{4}$	$\frac{\pi}{2}$	0.25	0.125
4	$\frac{7\pi}{4}$	$-\frac{\pi}{4}$	$\frac{\pi}{2}$	0.25	0.125

جدول۳: مشخصات هندسی چرخ‌ها

با توجه به رابطه‌ی ۱ که بیانگر رابطه‌ی محدودیت‌های اعمال شده توسط چرخ‌های ثابت و فرمان‌پذیر استاندارد به ربات می‌باشد و با توجه به داده‌ی موجود در جدول ۳ به محاسبه‌ی عناصر رابطه‌ی ۱ در معادلات ۲...۶ می‌پردازیم.

$$J_1 R(\theta)\dot{\xi}_I = J_2\dot{\phi} \tag{۱}$$

$$J_1 = \begin{bmatrix} \sin(\alpha_1+\beta_1+\gamma_1) & -\cos(\alpha_1+\beta_1+\gamma_1) & -l_1\cos(\beta_1+\gamma_1) \\ \sin(\alpha_2+\beta_2+\gamma_2) & -\cos(\alpha_2+\beta_2+\gamma_2) & -l_2\cos(\beta_2+\gamma_2) \\ \sin(\alpha_3+\beta_3+\gamma_3) & -\cos(\alpha_3+\beta_3+\gamma_3) & -l_3\cos(\beta_3+\gamma_3) \\ \sin(\alpha_4+\beta_4+\gamma_4) & -\cos(\alpha_4+\beta_4+\gamma_4) & -l_4\cos(\beta_4+\gamma_4) \end{bmatrix} = \begin{bmatrix} ۰ & -۱ & -۰/۱۷۶۸ \\ ۱ & -۰ & -۰/۱۷۶۸ \\ ۰ & +۱ & -۰/۱۷۶۸ \\ -۱ & +۰ & -۰/۱۷۶۸ \end{bmatrix} \tag{۲}$$

$$R(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & ۰ \\ -\sin(\theta) & \cos(\theta) & ۰ \\ ۰ & ۰ & ۱ \end{bmatrix} \tag{۳}$$

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^T \tag{۴}$$

$$J_2 = \begin{bmatrix} ۰/۱۲۵ & ۰ & ۰ & ۰ \\ ۰ & ۰/۱۲۵ & ۰ & ۰ \\ ۰ & ۰ & ۰/۱۲۵ & ۰ \\ ۰ & ۰ & ۰ & ۰/۱۲۵ \end{bmatrix} \tag{۵}$$

$$\dot{\phi} = \begin{bmatrix} \dot{\phi}_1 & \dot{\phi}_2 & \dot{\phi}_3 & \dot{\phi}_4 \end{bmatrix}^T \tag{۶}$$

که با انتقال معلومات معادله‌ی ۱ به سمت راست رابطه و جایگذاری معادلات ۲...۶ در معادله‌ی ۱ و اعمال

عملیات ریاضی مورد نیاز بروی مقادیر ثابت معادله، به معادله‌ی ۷ میرسیم.

$$\dot{\xi}_I(\phi_{۱\ldots۴},\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & ۰ \\ \sin(\theta) & \cos(\theta) & ۰ \\ ۰ & ۰ & ۱ \end{bmatrix} * \begin{bmatrix} ۰/۰۰۰۰ & +۰/۰۶۲۵ & +۰/۰۰۰۰ & -۰/۰۶۲۵ \\ -۰/۰۶۲۵ & -۰/۰۰۰۰ & +۰/۰۶۲۵ & -۰/۰۰۰۰ \\ -۰/۱۷۶۸ & -۰/۱۷۶۸ & -۰/۱۷۶۸ & -۰/۱۷۶۸ \end{bmatrix} * \begin{bmatrix} \dot{\phi}_۱ \\ \dot{\phi}_۲ \\ \dot{\phi}_۳ \\ \dot{\phi}_۴ \end{bmatrix} \quad (۷)$$

که معادله ی ۷ با جایگذاری زاویه‌ی کنونی‌ای که ربات دارد و سرعت تک‌تک چرخ‌ها و اعمال عملیات ضرب ها سرعت لحظه‌ای ربات در مختصات جهانی بدست خواهد آمد؛ و با انتگرال‌گیری از آن می‌توان موقعیت ربات را در مختصات جهانی بدست آورد.

۲.۳ درجه‌ی مانورپذیری ربات مدل‌سازی شده

با توجه به اینکه در مدل‌سازی نرم‌افزاری ربات برعکس نمونه واقعی ساخته شده؛ این مهم در نظر گرفته شده است که می شود به چرخ‌های مختلف سرعت‌های مختلف تخصیص داد(همان‌طور که معادله‌ی ۷ پیشنهاد میدهد) لازم دانستم که اندکی در مورد درجه‌ی مانورپذیری ربات بحثی کنم تا در مورد نتایج حاصل از شبیه‌ساز ربات(که در قسمت ۴ ارائه میشود) ابهامات احتمالی برطرف گردد.

همانطور که میدانیم درجه‌ی مانورپذیری ربات(δ_M) از رابطه‌ی ۸ بدست می آید که برابر با مجموع درجات فرمان‌پذیری و تحرک ربات است. از آنجایی که این ربات هیچ چرخ فرمان‌پذیری ندارد بنابراین درجه ی فرمان‌پذیری آن صفر بوده است و بدیهی است که درجه‌ی تحرک‌ای برابر با مقدار ۳ دارد؛ بنابراین درجه‌ی مانورپذیری برابر ۳ دارد و میتواند در هر موقعیتی خود را قرار دهد.

$$\delta_M = \delta_m + \delta_s \quad (۸)$$

۳.۳ کنترل ربات برای رسیدن به مختصات مشخص

همانطور که در قسمت ۲.۳ بحث شد این ربات میتواند با تنظیم مناسب چرخ‌های خود در هر موقعیتی خود را قرار دهد. کنترل پیاده‌سازی شده برای کنترل سرعت و درجه ربات از ۲ مرحله تشکیل شده است که ابتدا ربات با همان زاویه در حال حاضر قرار دارد خود را به موقعیت مورد نظر برساند و سپس در موقعیت هدف زاویه خود را تغییر دهد – در صورت انحراف از نقطه‌ی هدف در مرحله ۲ با استفاده از روند مرحله ی ۱ موقعیت خود را تصحیح کند سپس مرحله ۲ را ادامه دهد. که سرعت‌های هر یک از چرخ‌ها توسط معادله ۹ تخصیص داده میشود.

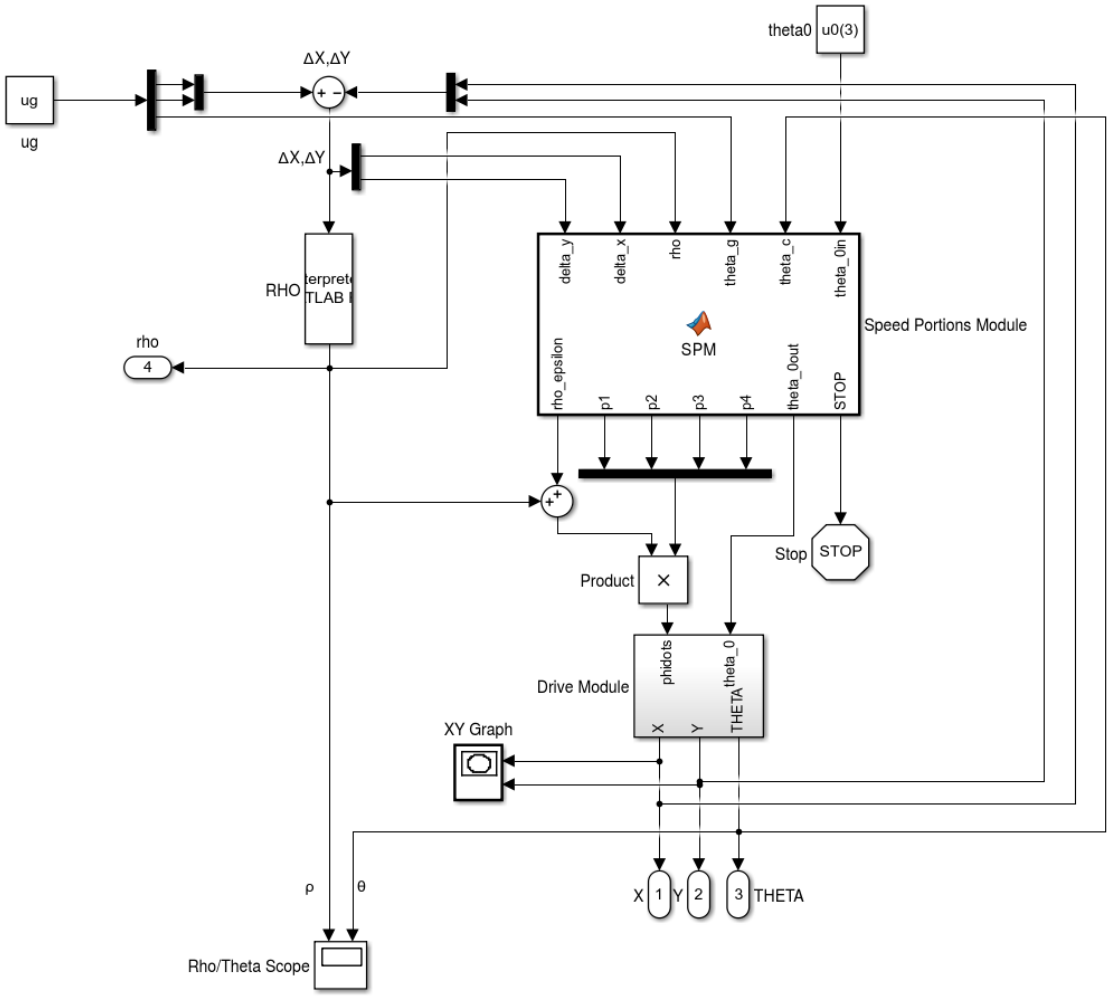
$$\begin{bmatrix} \dot{\phi}_۱' \\ \dot{\phi}_۲' \\ \dot{\phi}_۳' \\ \dot{\phi}_۴' \end{bmatrix} = \begin{bmatrix} -(K_\rho + K_\epsilon)\sin(\theta') \\ +(K_\rho + K_\epsilon)\cos(\theta') \\ +(K_\rho + K_\epsilon)\sin(\theta') \\ -(K_\rho + K_\epsilon)\cos(\theta') \end{bmatrix} \quad (۹)$$

که در معادله ۹ مقادیر K_ρ و θ' توسط ۱۰ و ۱۱ بدست می‌آیند.

$$K_\rho = \sqrt{\Delta x^2 + \Delta y^2} \qquad (10)$$

$$\theta' = \arctan 2(\Delta y, \Delta x) \qquad (11)$$

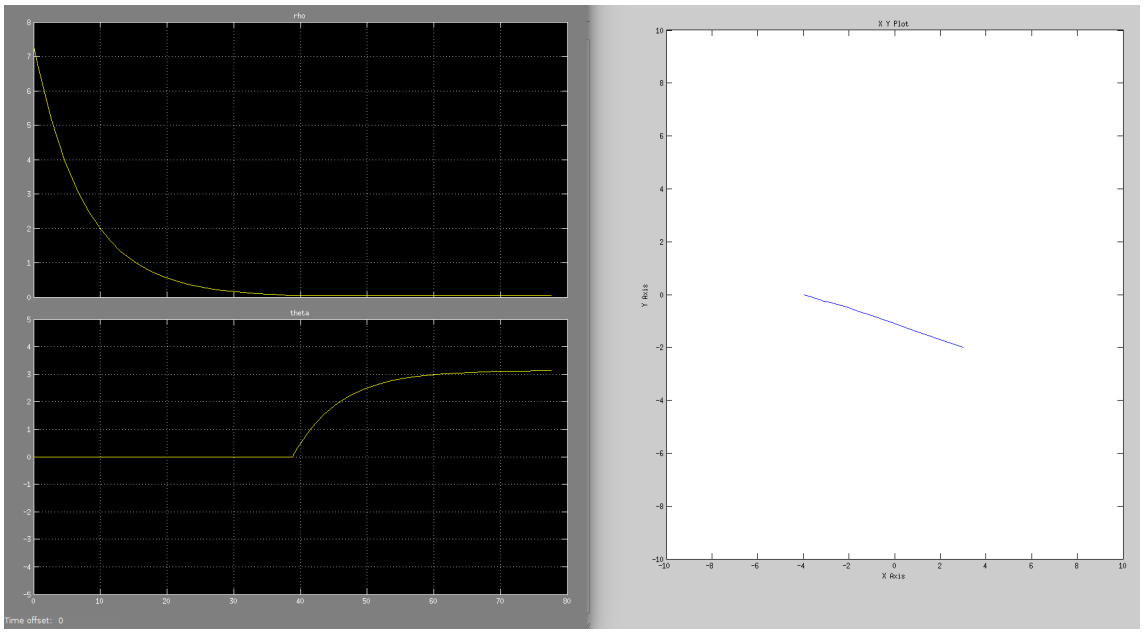
که در معادله ی ۹ عبارت K_ϵ برابر با یک مقدار دلخواه برای عادی‌سازی/بهینه‌سازی مقادیر K_ρ در برخی موارد خواص(معمولا مقداری برابر با صفر دارد) می باشد. و در معادلات ۱۰ و ۱۱ منظور از Δx و Δy فاصله‌ی مولفه های دکارتی موقعیت کنونی ربات با موقعیت هدف می‌باشند. که بلوک-دیاگرام سیمولینک شبیه‌ساز در شکل ۴ آورده شده است.



شکل ۴: بلوک-دیاگرام پیاده‌سازی شده در شبیه‌ساز

۴ نتیجه

در شکل ۵ نتیجه یک نمونه اجرا آورده شده است که در این نمونه اجرا از ربات خواسته شده است که از موقعیت $\begin{bmatrix} 3 & -2 & \frac{3\pi}{2} \end{bmatrix}^T$ *خود را به موقعیت* $\begin{bmatrix} -4 & 0 & \frac{\pi}{2} \end{bmatrix}^T$ *برساند.*



شکل۵: نتیجه یک نمونه اجرای بلوک‌ـدیاگرام شکل ۴ که از ربات خواسته شده است که موقعیت کنونی‌اش خود را به موقعیت هدفش برساند.

که همانطور که در شکل ۵ سمت راست میبینید ربات براحتی توانسته با یک مسیر مستقیم خود را به هدف برساند. در شکل ۵ سمت چپ ابتدا ربات به کاهش فاصله‌اش با هدف(چپ‌ـبالا) میپردازد پس از آنکه به موقعیت هدف رسید به تغییر زاویه‌ی خود(چپ‌ـپایین) به زاویه‌ی دلخواه می‌رساند؛ که شکل سمت چپ‌ـپایین میزان تغییرات زاویه انجام شده را نشان می‌دهد – که در ابتدا میزان تغییرات زاویه صفر بوده و سپس به اندازه $+\pi^\circ$ *تغییر زاویه داده است(همانطور که انتظار میرفت).*

۵ مراجع

[1] Roland Siegwart, IllahR. Nourbakhsh, and Davide Scaramuzza. Introduction to Autonomous Mobile Robots, *chapter3. The MIT Press, 2ed edition, 2011.*