

حل مسأله ی سینماتیک معکوس و کالیبراسیون دست به چشم بازوی ربات شش درجه آزادی سری با استفاده از شبکه های عصبی

آرمین غلامی، همایون مهدوی نسب، عباس چترایی

دانشکده ی مهندسی برق، واحد نجف آباد، دانشگاه آزاد اسلامی، نجف آباد، ایران

خلاصه

ربات های صنعتی، امروزه در بسیاری از صنایع، از قبیل صنایع نظامی، خودرو سازی و حتی مونتاژ برد های الکترونیکی کاربرد فراوان دارند و امروزه جزء جدایی ناپذیری از صنعت به حساب می آیند. طراحی کنترلر این ربات ها همیشه یکی از بزرگترین مشکلات طراحان بوده است. یکی از مسائل مهم در طراحی این کنترلر ها، به دست آوردن سینماتیک معکوس ربات ها می باشد. همچنین برای ارتباط داشتن بازو با محیط اطراف، طراحی بینایی مناسب نیز یکی از نیاز های اساسی طراحی این کنترلر ها به حساب می آید. لذا در این تحقیق، حل این دو مساله، به صورت همزمان به وسیله ی شبکه های عصبی مصنوعی مورد بررسی قرار گرفته و با شناسایی تصویر هدف و ارائه داده های آموزشی مناسب انجام شده است. نتایج ارائه شده دقت مناسب الگوریتم پیشنهادی را نشان می دهند. با استفاده از این روش، می توان کالیبراسیون دست به چشم و سینماتیک معکوس بازوهایی که در عمل، به دلیل پیچیدگی ساختار، از روش های معمول قابل محاسبه نیستند را به سادگی به دست آورد.

واژه های کلیدی: سینماتیک معکوس، کالیبراسیون دست به چشم، شبکه های عصبی مصنوعی، بینایی ماشین

۱. مقدمه

مساله ی سینماتیک معکوس با استفاده از روش های مختلفی از قبیل روش های تحلیلی کامل، روش های عددی و روش های نیمه تحلیلی مورد بررسی قرار گرفته است [۱]. در روش تحلیلی کامل تمام متغیر های مفصل را با توجه به داده های پیکر بندی داده شده بدست می آورند که برای مثال می توان به روش هندسی اشاره نمود. در روش دوم متغیر های مفصلی با استفاده از روشهای محاسباتی تکرار شونده بدست می آیند و در روش سوم برخی از متغیر ها به صورت تحلیلی و برخی به صورت عددی به دست خواهد آمد. محققان در سال های اخیر به دلیل طراحی بازوهای پیچیده تر و همچنین پیچیدگی های محاسباتی و کمبود های این روش ها، به روش های هوش مصنوعی از جمله الگوریتم های زنبور عسل [۲]، ژنتیک و ازدحام ذرات و منطق فازی و همچنین شبکه های عصبی مصنوعی [۳] و حتی ترکیب این روش ها [۴] روی آورده اند. مساله ی کالیبراسیون دست به چشم نیز، بسیاری از پژوهشگران را بیش از ۳۰ سال گذشته به خود جذب کرده است. این موضوع به طور سنتی با استفاده از روش مثلث بندی بررسی می شد که این روش نیازمند کالیبراسیون از فریم دوربین به فریم ضبط شده و یا فریم پایه ربات و مثلث بندی استریو است. لذا این روش بسیار وقت گیر و زمان بر است. همچنین این روش شامل پارامتر های درونی و بیرونی دوربین می باشد و عملکرد آن به شدت نسبت به این پارامترها حساس است.

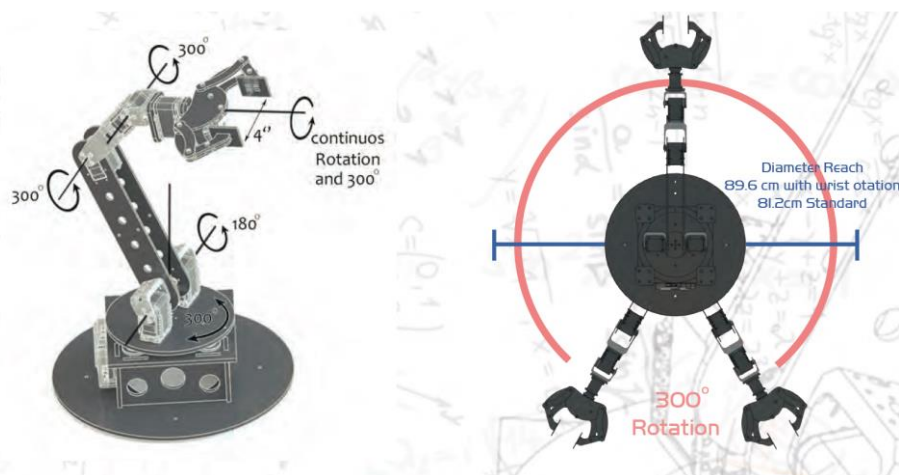
از جمله روش های دیگر حل این مساله می توان به الگوریتم های هوش مصنوعی، به عنوان مثال الگوریتم ازدحام ذرات [۵] و همچنین شبکه های عصبی [۶ - ۷] اشاره نمود. به وسیله ی شبکه های عصبی، می توان راه حلی پیشنهاد کرد که بدون نیاز به این پارامتر ها، این مساله با دقت قابل قبولی حل شود.

در روش شبکه های عصبی، از شبکه های مختلف از قبیل شبکه ی پرسپترون چند لایه [۸]، شبکه ی عصبی RBF [۹ - ۱۰]، شبکه های عصبی با معماری مدولار [۱۱ - ۱۲]، شبکه های عصبی LRNNs [۱۳]، شبکه های عصبی چهار گانه [۱۴] و همچنین توپولوژی های مختلف این شبکه ها استفاده شده است. در بعضی از این مقالات بازوهای با افست نیز مورد بررسی قرار گرفته اند [۱۵].

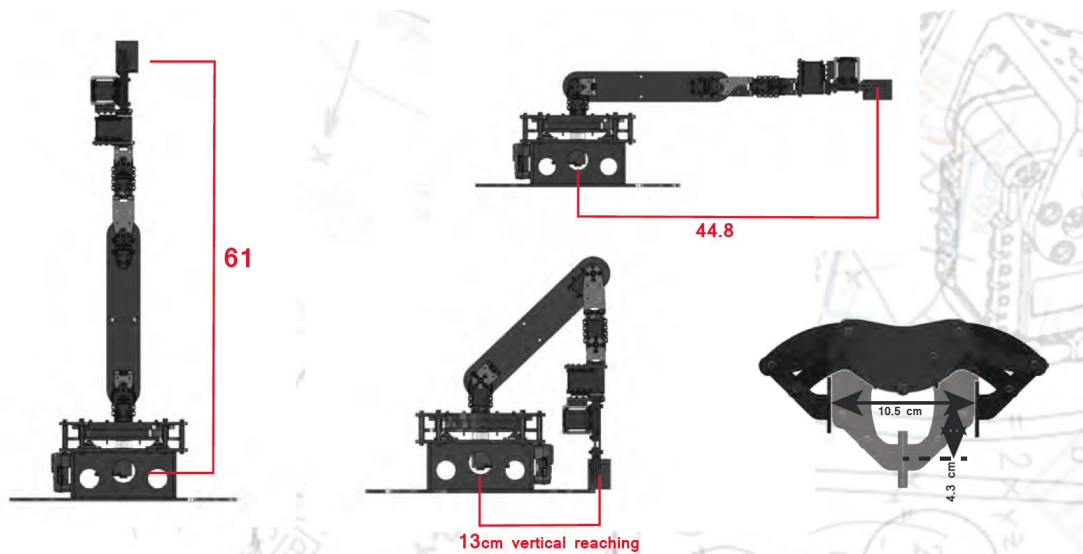
در تحقیقاتی که سینماتیک معکوس به تنهایی بررسی می شد، به دلیل ضعف های موجود در طراحی و خطاهای محاسباتی و هماهنگ نبودن دوربین با بازو، خطا مقدار قابل توجهی داشت. لذا در این تحقیق، سعی بر این است که با ترکیب سینماتیک معکوس بازوی ربات با کالیبراسیون دست به چشم، هر دو مساله ی بالا به طور همزمان بررسی شوند و مقدار خطا کمینه شود.

۲. مدل سیستم

مدل استفاده شده در این پژوهش، یک بازوی ربات سری شش درجه آزادی (با احتساب گریپر)، با نام ARM6AX18، ساخته شده از ورقه های پلکسی گلس ۳ میلیمتری می باشد (شکل ۱). سیستم محرکه ی این بازو متشکل از هفت عدد سروو موتور داینامیکسل می باشد که وظیفه ی جا به جایی لینک های ربات را بر عهده دارند. این موتور ها قابلیت چرخش بین ۰ تا ۳۰۰ درجه را دارا می باشند و برای حرکت در این زوایا مقادیر ۰ تا ۱۰۲۳ به وسیله ی ارتباط سریال از طریق کامپیوتر یا کنترل کننده به موتور ارسال می شود. لذا دقت هر یک از موتور ها حدود ۰٫۳ درجه می باشد. وزن بازو حدودا ۱۶۸۰ گرم، حداکثر فاصله قابل دسترس افقی ۴۴٫۸ سانتیمتر، فاصله پایه تا محور شانه ۱۲٫۱ سانتیمتر، فاصله محور شانه تا محور آرنج ۲۰٫۲ سانتیمتر و فاصله محور آرنج تا محور مچ ۱۶٫۸ سانتیمتر می باشد (شکل ۲). طول دسترسی بازو در حالت افقی ۴۴٫۸ سانتیمتر و در حالت عمودی ۶۱ سانتیمتر است. همچنین این بازو در طول ۱۳ سانتیمتر توانایی بلند نمودن ۷۰۰ گرم بار و در طول ۳۰ سانتیمتر ۲۲۵ گرم بار را دارا می باشد. گریپر توانایی نگه داشتن بار اسمی تا ۴۰۰ گرم را دارد. مچ این بازو به تنهایی می تواند ۲۴۰ گرم بار را بلند کند.



شکل ۱- محدوده ی حرکت موتور های هر لینک



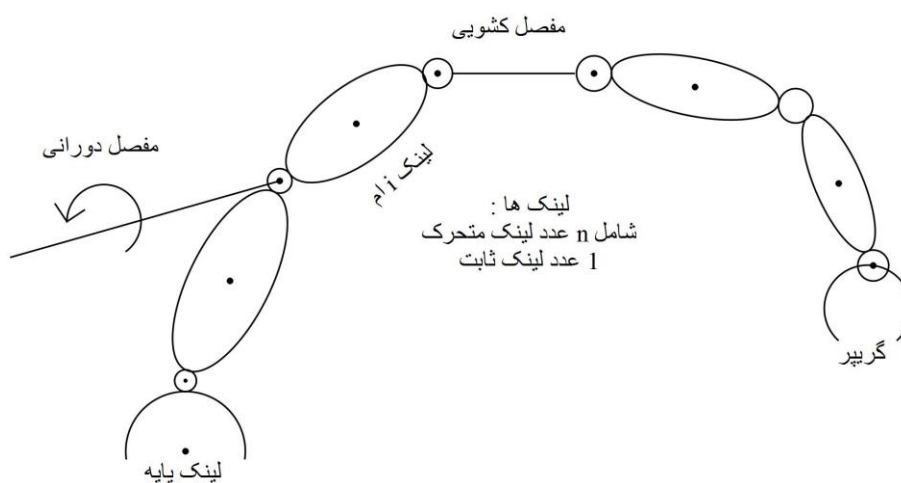
شکل ۲- محدوده ی دسترسی بازوی ربات

۳. سینماتیک مستقیم ربات

به طور کلی به مطالعه ی موقعیت و جهت گیری مجری نهایی بازو، با معلوم بودن مقادیر و متغیر های مفصلی، سینماتیک مستقیم گفته می شود.

بازوهای سری از مجموعه ای از لینک ها که با مفاصل به هم متصل می شوند، تشکیل شده است (شکل ۳). مقادیر یا متغیر های مفصلی با توجه به نوع مفصل، به انواع مختلفی تقسیم بندی می شوند. این مفصل ها، به مفصل های ساده، مانند مفصل های دورانی یا کشویی، و مفصل های پیچیده، مانند مفصل گوی و ساچمه تقسیم بندی می شوند. مفصل های ساده دارای یک درجه آزادی هستند، در حالی که درجه آزادی مفصل های پیچیده بیشتر از یک می باشد.

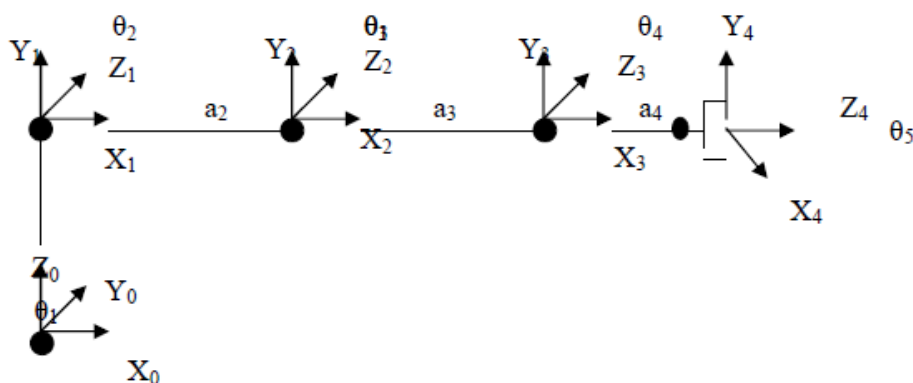
بازوی ربات



شکل ۳- شماتیک یک بازوی سری متشکل از مفصل های کشویی و دورانی

۴. محاسبه ی سینماتیک مستقیم ربات

بازوی ARM6AX18 یک بازوی پنج درجه آزادی به همراه یک گریپر می باشد. همچنین با این تعداد مفصل از لحاظ عملکرد بسیار به بازوی انسان شبیه می باشد. این مفصل ها چرخش شانه، حرکت جلو به عقب شانه، حرکت آرنج، حرکت بالا و پایین مچ، چرخش مچ و حرکت گریپر را مهیا می کند. در شکل (۴) نمای گرافیکی این بازو را مشاهده می کنید.



شکل ۴- نمای گرافیکی بازوی ربات پنج درجه آزادی ARM6AX18

تمامی مفصل این بازو از نوع دورانی می باشند. با استفاده از الگوریتم دناویت-هارتنبرگ و یافتن چارچوب های پایه مانند شکل (۴)، جدول آن به صورت زیر بدست می آید :

Joint	θ_i	α_i	a_i	d_i
1	θ_1	90	0	0
2	θ_2	0	a_2	0
3	θ_3	0	a_3	0
4	θ_4	-90	a_4	0
5	θ_5	90	0	0
6	θ_6	0	0	0

جدول ۱ - پارامتر های دناویت هارتنبرگ بازوی ARM6AX18

لذا ماتریس های انتقال A_1 تا A_6 به صورت زیر محاسبه می شوند :

$$T_2^1 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2 C_2 \\ S_2 & C_2 & 0 & a_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_1^0 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^2 = \begin{bmatrix} C_3 & -S_3 & 0 & a_3 C_3 \\ S_3 & C_3 & 0 & a_3 S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_5^4 = \begin{bmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_6^5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} T_4^3 = \begin{bmatrix} C_4 & 0 & -S_4 & a_4 C_4 \\ S_4 & 0 & C_4 & a_4 S_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ماتریس تبدیل (T) از حاصل ضرب شش ماتریس بالا به دست می آید. با محاسبه ی این ماتریس موقعیت و جهت گیری ربات در هر نقطه مشخص می شود.

$$= \begin{bmatrix} R_6^0 & P_6^0 \\ 0_3^T & 1 \end{bmatrix} T_6^0 = T_1^0 \times T_2^1 \times T_3^2 \times T_4^3 \times T_5^4 \times T_6^5 \quad (1)$$

$$= \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$n_x = C_1 C_{234} - C_5 S_1 S_5 \quad (2)$$

$$n_y = S_1 C_{234} C_5 + C_1 S_5 \quad (3)$$

$$n_z = S_{234} C_5 \quad (4)$$

$$o_x = -C_1 S_{234} \quad (5)$$

$$o_y = -S_1 S_{234} \quad (6)$$

$$o_z = C_{234} \quad (7)$$

$$a_x = C_1 C_{234} S_5 + S_1 C_5 \quad (8)$$

$$a_y = S_1 C_{234} S_5 - C_1 C_5 \quad (9)$$

$$a_z = S_{234} S_5 \quad (10)$$

$$(11)$$

$$P_x = C_1 (C_{23} a_3 + C_2 a_2 + C_{234} a_4) \quad (12)$$

$$P_y = S_1 (C_{23} a_3 + C_2 a_2 + C_{234} a_4) \quad (13)$$

$$P_z = S_{23} a_3 + S_2 a_2 + S_{234} a_4$$

۵. تهیه ی نمونه های اولیه

نمونه های اولیه شامل لیستی از زاویه های ۴ مفصل اولیه ربات می باشد. لازم به ذکر است که معمولا برای دسترسی به هر نقطه از فضا با جهت گیری های دلخواه، به بازویی با شش درجه آزادی نیاز داریم. ولی بازوی ربات ARM6AX18 تنها دارای پنج درجه آزادی می باشد. لذا تنها در صفحه دارای جهت گیری های مختلف است. نکته دیگر این است که این بازو به دلیل استفاده از سروو موتور و در نتیجه مقادیر گسسته زوایای موتور هایش، توانایی رفتن به همه ی جهت گیری های قابل محاسبه را نیز ندارد. لذا در این پژوهش تنها موقعیت ربات مورد بررسی قرار می گیرد. به دلیل اینکه زاویه ی مفصل پنجم هیچ نقشی در موقعیت نوک ربات ندارد، لذا داده های اولیه شامل زاویه ی مفصل پنجم نمی شود. داده های اولیه در یک مختصات کروی به چهار قسمت تقسیم شدند.

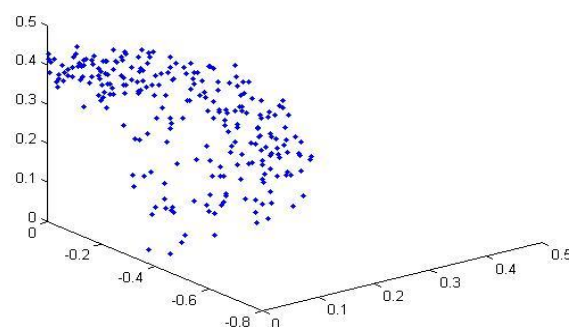
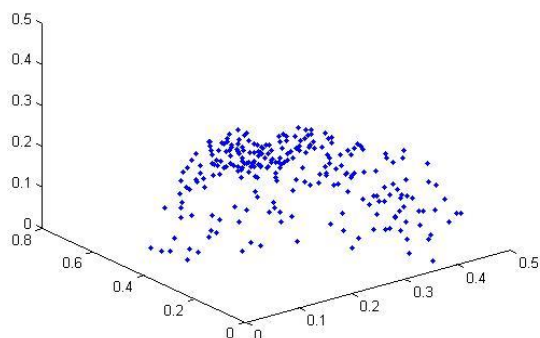
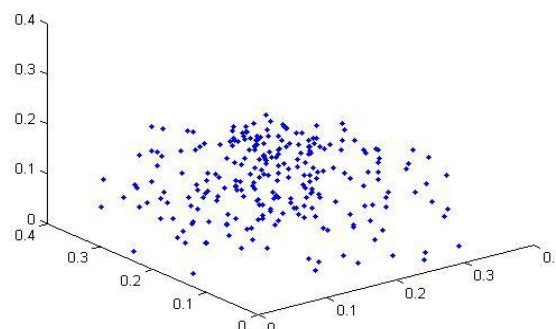
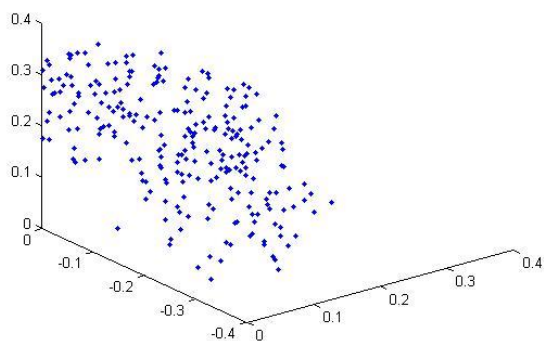
$$0 \leq \theta \leq P_i/2, \quad 0.14 \leq r \leq 0.35, \quad 0 \leq \varphi \leq P_i/2$$

$$pi/2 \leq \theta \leq pi, \quad 0.14 \leq r \leq 0.35, \quad 0 \leq \varphi \leq P_i/2$$

$$0 \leq \theta \leq P_i/2, \quad 0.35 < r \leq 0.44, \quad 0 \leq \varphi \leq P_i/2$$

$$pi/2 \leq \theta \leq pi, \quad 0.28 < r \leq 0.46, \quad 0 \leq \varphi \leq P_i/2$$

برای جمع آوری این داده ها، ابتدا با نرم افزار متلب یک ماتریس با مقادیر تصادفی صحیح با توزیع یکنواخت ایجاد شد، به صورتی که این مقادیر در محدوده ی کار موتور ها باشند. سپس این داده ها به برنامه ی سینماتیک مستقیم ربات اعمال شد و موقعیت های گریپر ربات به ازای این داده ها محاسبه شد. سپس با تبدیل این مختصات دکارتی به مختصات کروی و اعمال شرایط فوق، مقادیر خارج از محدوده حذف و داده ها به چهار قسمت تقسیم شدند (شکل ۵).



شکل ۵- چهار ناحیه متشکل از مختصات نوک ربات، که بعد از اعمال داده های اولیه به سینماتیک مستقیم ربات بدست آمده است

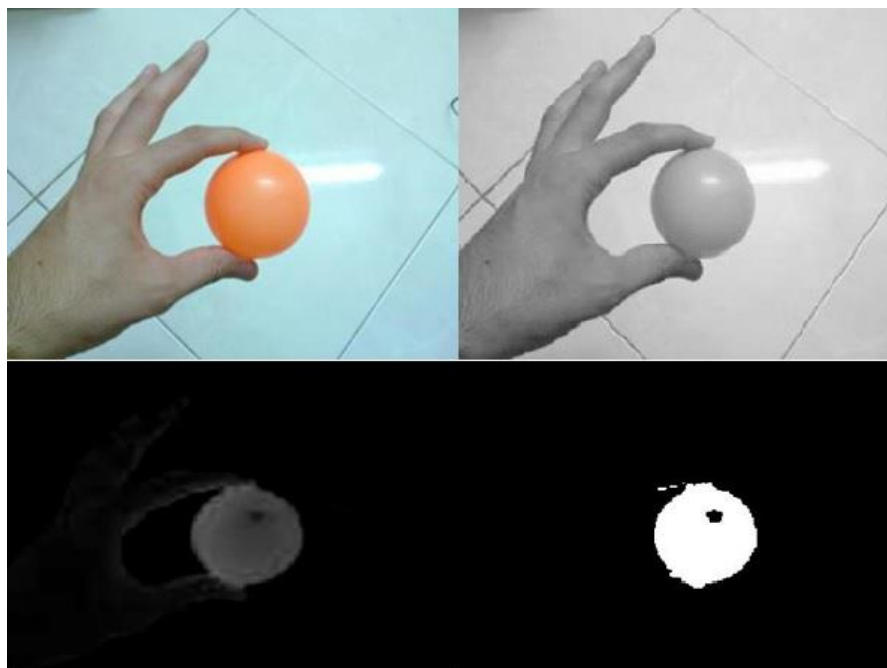
۶. پردازش تصویر جهت شناسایی هدف

برای سیستم بینایی از دو عدد و بکم و دو عدد پایه استفاده شده است. این دو دوربین طوری استقرار یافته اند که محدوده ی نمونه برداری ربات را به طور کامل پوشش دهند و تا حد امکان هیچ مانعی حتی بدنه ی ربات، مانع دید دوربین ها نشود.

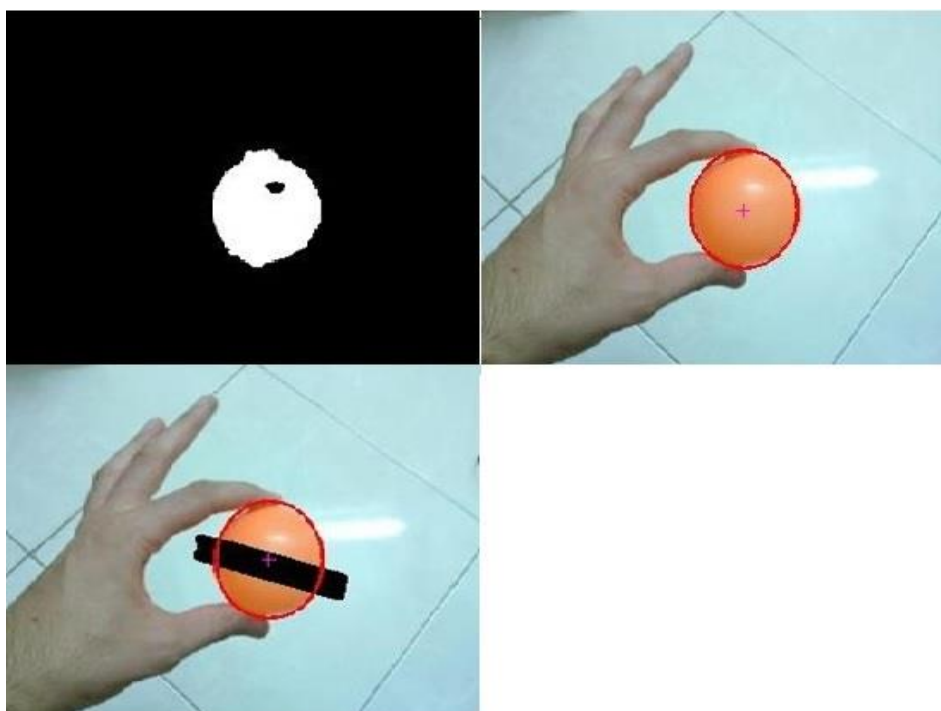
هدف، پیدا کردن مختصات دو بعدی مرکز یک توپ نارنجی در تصویر های این دو دوربین می باشد. الگوریتم های پردازش تصویر بسیار زیادی برای این کار وجود دارد، ولی به دلیل اینکه این کار صرفا به منظور تهیه ی داده های مورد نیاز برای آموزش شبکه ی عصبی مورد نظر می باشد و پشت صحنه ی این تصاویر تا حد امکان ساده انتخاب شده است، و همچنین برای بالا بردن سرعت پردازش و تهیه ی داده ها، از الگوریتم های ساده، اما دقیق پردازش تصویر استفاده می شود. به منظور پیاده سازی این الگوریتم از جعبه ابزار پردازش تصویر متلب استفاده شده است.

الگوریتم پردازش تصویر به شرح زیر می باشد :

- (۱) وارد کردن تصویر به محیط نرم افزار
- (۲) تبدیل تصویر به تصویر سیاه و سفید
- (۳) کم کردن ماتریس مقادیر قرمز تصویر، از تصویر سیاه سفید
- (۴) باینری کردن تصویر با سطح خاکستری ۰,۱۸
- (۵) حذف نویز های موجود در تصویر : نحوه ی انجام این کار به این صورت می باشد که تمامی اشیائی که دارای مساحت کمتر از ۳۰۰ پیکسل هستند را حذف می کنیم.
- (۶) نشانه گذاری نواحی مختلف موجود در تصویر
- (۷) به دست آوردن مختصات و مشخصات نواحی مختلف
- (۸) مساله ی مهم در اینجا این است که ممکن است در موقعیت های خاصی گریپر بازو جلوی توپ بیاید. در این وضعیت تعداد نواحی تصویر بیشتر از ۱ ناحیه می شود. لذا برای عملکرد درست برنامه در این مرحله می توانیم میانگین مختصات نواحی را به عنوان مختصات مرکز توپ بدست آوریم.



شکل ۶- مراحل ۱ تا ۴ الگوریتم پردازش تصویر



شکل ۷- مراحل ۵ تا ۸ الگوریتم پردازش تصویر

۷. تهیه ی داده های ورودی شبکه ی عصبی

تا این مرحله، لیستی از زوایای موتور های ربات، که همان خروجی های شبکه ی عصبی می باشند، تهیه کرده ایم. این زوایا به چهار ناحیه تقسیم بندی شدند. این تقسیم بندی بیشتر به دلیل ضعف حرکت ربات بین دو نقطه نزدیک به هم انجام شد. در این مرحله، هدف اعمال این زوایا به موتور های بازوی ربات و تهیه ی دو لیست، متشکل از مختصات مرکز توپ در تصویر هر یک از دوربین ها می باشد. به این منظور، ابتدا باید محل قرار گیری دوربین ها مشخص شود.

دوربین ها، با فاصله ی حدودا ۱,۵ متر از چارچوب مرجع بازو قرار گرفتند. فاصله ی عمودی فریم دوربین ها با فریم مرجع بازو، ۳۰ سانتی متر می باشد. بر خلاف روش مثلث بندی، که باید زاویه دید دوربین ها با هم موازی باشند، زاویه دید دوربین ها در امتداد بازو می باشد.

این نکته قابل توجه است که چینش دوربین ها با فواصل و زوایای دقیق انجام نشده است و مقادیر ذکر شده، به صورت کاملا تقریبی می باشد. بر خلاف روش مثلث بندی، که چینش دوربین ها طبق اصول و قواعد خاص انجام می شود و کوچکترین اختلافی بین مقدار تئوری آنها با مقادیر عملی، باعث به وجود آمدن خطاهای بزرگ می شود، در روش شبکه عصبی، چینش دوربین ها می تواند کاملا به صورت تصادفی باشد و حتی نیازی به تراز کردن دوربین ها نمی باشد. تنها مساله ای که باید مورد توجه قرار بگیرد این است که، دوربین ها باید به نحوی قرار بگیرند که کل ناحیه ی نمونه گیری در محدوده ی دید آنها باشند.

بعد از آماده سازی سیستم بینایی، لیست زوایای موتور ها، یکی یکی به ربات اعمال شد و پس از حرکت ربات به مقصد مورد نظر و تهیه ی تصویر، مختصات دو بعدی توپ از دو تصویر استخراج شد (شکل ۸). مشکل ابتدایی در استخراج این

مختصات، قرار گرفتن گریپر، جلوی دید دوربین بود. لذا در مختصات هایی که گریپر جلوی دوربین قرار می گرفت، با چرخش مناسب گریپر، توپ در دید مستقیم دوربین قرار می گرفت. با این روش، این مشکل تا حد مطلوبی رفع گردید.

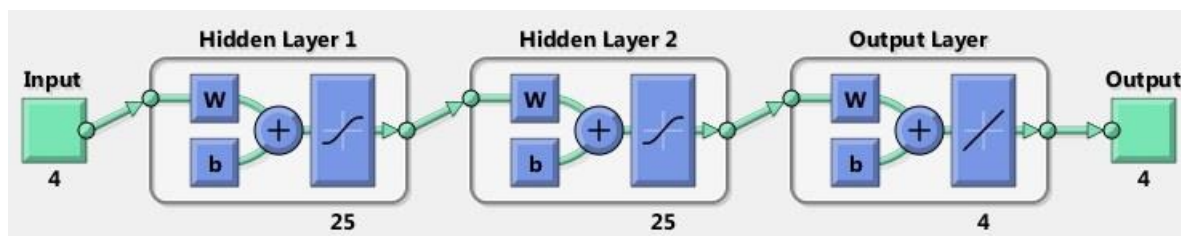


شکل ۸- تعیین مختصات مرکز توپ در تصویر دوربین اول و دوم

۸. آموزش شبکه ی عصبی

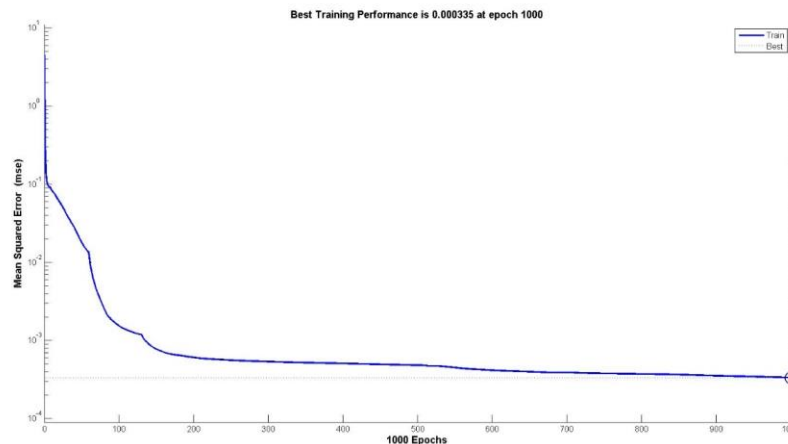
در مرحله ی قبل، کلیه ی داده های مورد نیاز برای آموزش شبکه ی عصبی تهیه گردید. قدم بعدی، انتخاب توپولوژی مناسب شبکه، شامل تعداد لایه های مخفی، تعداد نرون ها و تابع فعالیت هر لایه می باشد.

با تست توپولوژی های مختلف، شامل تعداد لایه های مخفی، تعداد نرون های هر لایه و همچنین تابع فعالیت هر لایه، شبکه ی نشان داده شده در شکل (۹) به عنوان شبکه ی منتخب برای این سیستم در نظر گرفته شد. این شبکه با چهار ورودی و چهار خروجی، شامل دو لایه ی مخفی، با تابع فعالیت Tansig و لایه ی خروجی با تابع فعالیت خطی می باشد. تعداد نرون های هر کدام از لایه های مخفی ۲۵ عدد و لایه ی خروجی ۴ عدد می باشند.



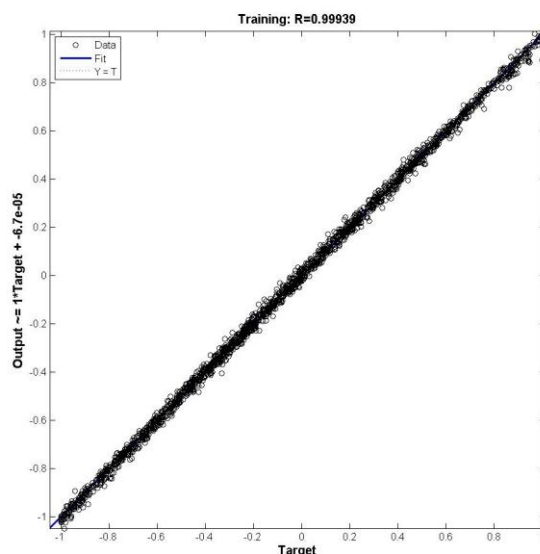
شکل ۹- توپولوژی شبکه ی عصبی پیشنهادی

به منظور آموزش این شبکه با داده های موجود، ابتدا مقادیر وزن ها و بایاس ها، مقدار دهی اولیه می شوند. سپس آموزش به نحوی انجام می شود که تابع کارایی شبکه حداقل شود. همانطور که در شکل (۱۰) مشاهده می شود، پس از ۱۰۰۰ تکرار، شبکه ی عصبی با شاخص عملکرد ۰,۰۰۰۳۳۵ آموزش داده شد.



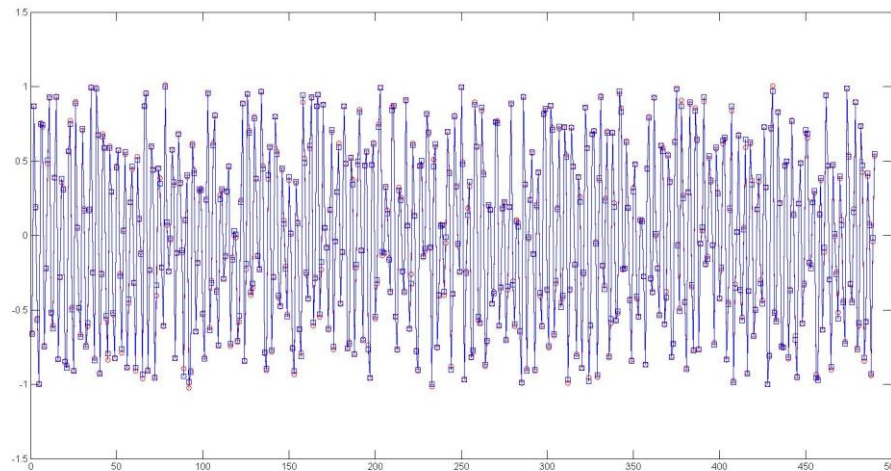
شکل ۱۰- کاهش تابع کارایی در طول آموزش شبکه

نمودار رگرسیون این آموزش در شکل (۱۱) آورده شده است. همانگونه که مشاهده می شود، خروجی شبکه ی عصبی، بسیار نزدیک به خط $X=Y$ می باشد که آموزش مناسب شبکه ی عصبی را نشان می دهد.



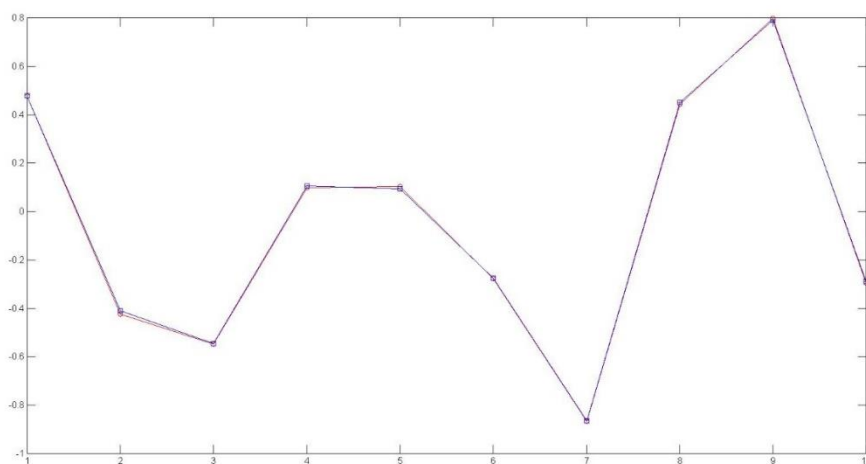
شکل ۱۱- نمودار رگرسیون آموزش شبکه ی عصبی

به منظور بررسی بهتر عملکرد این شبکه، مقادیر زوایای بدست آمده از مرحله ی نمونه برداری از یک لینک ربات، با مقادیر خروجی شبکه ی عصبی برای آن لینک در شکل (۱۲) مشاهده می شود. این نمودار بیانگر این مطلب است که داده های آموزش شبکه ی عصبی به خوبی آموزش داده شده اند و خطای آموزش بسیار ناچیز می باشد. قابل ذکر است که مقادیر قابل مشاهده در این شکل، نرمالایز شده هستند، لذا کلیه ی مقادیر در محدوده ی $[-1, 1]$ می باشند.



شکل ۱۲- مقایسه ی داده های اصلی و خروجی شبکه ی عصبی

در قدم بعدی، به منظور بررسی اعتبار سنجی و تست این شبکه ی عصبی، تعداد ۱۰ عدد از داده هایی که در آموزش شبکه استفاده نشده بودند، به شبکه اعمال شدند و مانند قسمت قبل، برای نمونه، داده های یک لینک در شکل (۱۳) نشان داده شده اند. قابل ذکر است که مقادیر آبی رنگ، مقادیر بدست آمده در مرحله ی نمونه گیری و مقادیر قرمز رنگ، مقادیر خروجی شبکه ی عصبی می باشند. همانگونه که مشاهده می شود، این داده ها تقریباً بر هم منطبق می باشند.



شکل ۱۳- داده های تست اعمال شده به شبکه ی عصبی

۹. نتیجه گیری

همانگونه که می دانید، خطاهای موجود در یک سیستم بسیار وسیع می باشند. از جمله خطاهای موجود در سیستم مورد بحث، می توان به خطاهای اندازه گیری، خطای ساخت و مونتاژ قطعات، خطای مربوط به صلب نبودن لینک های ربات، خطای مربوط به دقت موتور های استفاده شده در ربات، خطاهای مربوط به سیستم بینایی، شامل خطاهای ناشی از

نور پردازی، نوع دوربین های استفاده شده در یک سیستم، و خطاهای مربوط به دقت الگوریتم های مختلف استفاده شده برای پردازش اطلاعات و تصاویر را می توان نام برد. با توجه به عملکرد و نحوه ی پیشرفت این پروژه، بسیاری از خطاهای فوق به مقدار چشم گیری کاهش خواهند یافت. همچنین این روش بر خلاف روش های قبل، نیازی به شرط های مختلف چینش دوربین ها و تنظیم دقیق فواصل ندارد. با این حال، خطاهایی از قبیل دقت سروو موتور ها، نورپردازی سیستم و همچنین کیفیت و دقت دوربین ها، در خطای نهایی سیستم نقش دارند. لذا نورپردازی صحیح و استفاده از تجهیزات مناسب، دقت نهایی این روش را به مراتب بالاتر خواهد برد.

یکی از مسائل اساسی در بدست آوردن سینماتیک معکوس بازو، که مانع از طراحی بسیاری از بازوهای مختلف می شود، ناتوانی در به دست آوردن سینماتیک معکوس آن بازو و یا نبودن الگوریتم معینی که بتوان سینماتیک معکوس را با هر ساختاری به دست آورد، می باشد. با روش ارائه شده در این تحقیق، می توان سینماتیک معکوس هر بازو، با هر ساختار و هر تعداد درجه ی آزادی را به دست آورد.

مراجع

1. J. Craig, Introduction to Robotics: Mechanics and Control, Pearson Education, 2005.
2. D. Pham, M. Castellani and A. Fahmy, "Learning the inverse kinematics of a robot manipulator using the Bees Algorithm," in *IEEE, Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on*, Daejeon, 2008.
3. S. Tejomurtula and S. K. B., "Inverse kinematics in robotics using neural networks," *ELSEVIER, information sciences*, vol. 116, pp. 147-164, 1999.
4. L.-X. Wei, H.-R. Wang and Y. Li, "A new solution for inverse kinematics of manipulator based on neural network," *IEEE, Machine Learning and Cybernetics, 2003 International Conference on*, vol. 2, pp. 1201 - 1203, 2003.
5. A.Li, Z.Ma, Y.Hu and N.Lin, "Hand-eye calibration of measurement robot based on multi-population particle swarm optimization," in *IEEE, Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, Chongqing, 2008.
6. J. Jun, C.Kim, "Robust camera calibration using neural network," in *IEEE, TENCN 99. Proceedings of the IEEE Region 10 Conference*, Cheju Island, 1999.
7. M. Jones and D. Vernon, "Using neural networks to learn hand-eye co-ordination," *Neural Computing & Applications*, vol. 2, no. 1, pp. 2-12, 1994.
8. Y. Feng, W. Yao-nan and Y. Yi-min, "Inverse Kinematics Solution for Robot Manipulator based on Neural Network under Joint Subspace," *INT J COMPUT COMMUN*, vol. 7, pp. 459-472, 2012.

9. Y. Yang, G. Peng, Y. Wang and H. Zhang, "A New Solution for Inverse Kinematics of 7-DOF Manipulator Based on Neural Network," in *IEEE, Automation and Logistics, 2007 IEEE International Conference on* , Jinan, 2007.
10. Y. Ming, L. Jiangeng and L. Guizhang, "The inverse kinematics control algorithm based on RBF neural networks for manipulators," *IEEE, Intelligent Control and Automation, 2004. WCICA 2004. Fifth World Congress on*, vol. 6, no. Intelligent Control and Automation, pp. 4951 - 4955, 2004.
11. E. OYAMA and N. Y. Chong, "Inverse Kinematics Learning by Modular Architecture Neural Networks with Performance Prediction Networks," *IEEE, Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on* , vol. 1, pp. 1006-1012, 2001.
12. E. Oyama, A. Agah, K. F. MacDorman, T. Maeda and S. Tachi, "A modular neural network architecture for inverse kinematics model learning," *ELSEVIER, Neurocomputing* , vol. 38, pp. 797-805, 2001.
13. Y. Al-Mashhadany, "Inverse Kinematics Problem (IKP) of 6-DOF Manipulator by Locally Recurrent Neural Networks (LRNNs)," in *IEEE, Management and Service Science (MASS), 2010 International Conference on* , Wuhan, 2010.
14. Y. Cui, K. Takahashi and M. Hashimoto, "Design of control systems using quaternion neural network and its application to inverse kinematics of robot manipulator," in *IEEE, System Integration (SII), 2013 IEEE/SICE International Symposium on* , Kobe, 2013.
15. Z. Bingul, H. Ertunc and C. Oysu, "Comparison of inverse kinematics solutions using neural network for 6R robot manipulator with offset," in *IEEE, Computational Intelligence Methods and Applications, 2005 ICSC Congress on*, Istanbul, 2005.