

کنترل خودفرمان بینایی بازوی ربات در فضای سه بعدی با سیستم‌های فازی و مدل‌سازی عصبی

فرزانه نادی^۱، ولی درهمی^{۲*}

۱- کارشناسی ارشد دانشکده مهندسی کامپیوتر، پردیس فنی و مهندسی، دانشگاه یزد، ایران farzane.nadi@stu.yazd.ac.ir
 ۲* نویسنده مسئول: دانشیار، دانشکده مهندسی کامپیوتر، پردیس فنی و مهندسی، دانشگاه یزد، ایران vderhami@yazd.ac.ir

چکیده- سیستم کنترل خودفرمان بینایی، به سیستمی اطلاق می‌شود که از اطلاعات بازخوردی دوربین برای کنترل ربات استفاده می‌کند؛ تا ربات، از نقاط شروع دلخواه به نقطه هدف برسد. با توجه به نیاز به کنترل ربات در فضای سه بعدی، مختصات نقاط در سه بعد مورد نیاز است. بدین منظور در این مقاله، ایده استفاده از دوربین کینکت که علاوه بر تصویر رنگی، ماتریس عمق محیط را هم می‌دهد؛ دنبال شده است. فرمان کنترل با استفاده از ماتریس ژاکوبین تولید می‌شود. از آنجا که مدلی ریاضی از ترکیب ربات و دوربین در دسترس نیست، لذا با جمع آوری داده‌های مناسب با استفاده از شبکه‌های عصبی مصنوعی، معکوس ماتریس ژاکوبین تقریب زده می‌شود؛ تا مستقیماً در قانون کنترل مورد نظر استفاده شود. به ازای هر کدام از درجات آزادی بازوی ربات، یک شبکه عصبی دو لایه با ساختار پیشرو در نظر گرفته شده است. فاصله مجری نهایی با هدف در سه بعد و زاویه مفصل شانه، ورودی‌های این شبکه‌ها هستند؛ و خروجی آنها، بیان‌کننده میزان تغییرات ویژگی‌ها به تغییر در مقادیر متغیرهای مفصل ربات (المان‌های معکوس ماتریس ژاکوبین) می‌باشد. در قانون کنترل، از ماتریس قطری بهره استفاده می‌شود. درایه‌های روی قطر این ماتریس توسط سیستم خبره فازی ارائه شده تعیین می‌گردند. روش ارائه شده بر روی یک بازوی ربات صنعتی پیاده‌سازی شده است. نتایج آزمایشات حاکی از موفقیت روش پیشنهادی در رساندن مجری نهایی به نقاط هدف مختلف در محیط کاری با دقت مناسب و تعداد گام کمتر نسبت به روش پیشین است. واژه‌های کلیدی: کنترل خودفرمان بینایی، ماتریس ژاکوبین، سیستم فازی، شبکه‌های عصبی مصنوعی.

۱- مقدمه

کنترل کند [۱]. این روش، راه‌حلی برای کنترل ربات در محیط‌های ساخت یافته می‌باشد و شامل تکنیک‌های مختلفی از جمله پردازش تصویر، بینایی ماشین و تئوری کنترل می‌باشد [۲-۴]. از اجزای مهم سیستم‌های رباتیکی جمع‌آوری، پردازش و تفسیر داده‌های حسگرها می‌باشد [۴] و بنا بر طراحی مورد نظر، حسگرها می‌تواند در یک مکان به صورت ثابت نصب شود^۲ و یا روی بازوی ربات قرار بگیرد^۳ [۱، ۵]. پردازش اصلی در کنترل خودفرمان بینایی، همانطور که در شکل (۱) مشخص می‌باشد، به گونه‌ای است که واحد تخمین حالت با استفاده از بردار حالت سیستم (X)، بردار اندازه (Y) که همان مقادیر ویژگی‌ها می‌باشد، را محاسبه می‌کند. سپس واحد کنترل با استفاده از بردار اندازه و بردار اندازه مطلوب (Y*) حرکت نسبی اقلیدسی ربات را به عنوان

سال‌های بسیاری است که در کشورهای صنعتی استفاده از ربات گسترش زیادی داشته و عمده استفاده آن برای جابجایی از یک نقطه به نقطه مطلوب می‌باشد. لذا محققین سیستم‌های هوشمند و رباتیک در پی ارائه راه‌حل‌های کارا و ارزان قیمت برای کنترل بازوهای رباتیک بوده‌اند. از آنجا که دوربین اطلاعاتی با دقت مناسب، منعطف، و در عین حال کم هزینه را در دسترس قرار می‌دهد، به عنوان پرکاربردترین حسگر مورد استفاده در سیستم‌های رباتیکی می‌باشد. به این مکانیزم کنترل ربات برای رسیدن به هدف، کنترل خودفرمان بینایی^۱ گفته می‌شود. کنترل خودفرمان بینایی تکنیکی است که از داده‌های بازخوردی استخراج شده توسط حسگرهای بصری استفاده می‌کند تا حرکت ربات را

² Fixed Camera

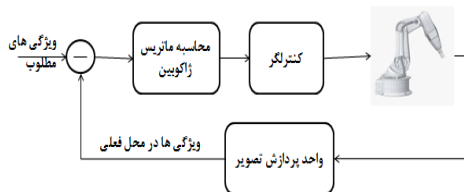
³ Eye-in-hand

¹ Visual Servoing

مدل از ربات، کاری دشوار و زمانبر خواهد بود و روش‌های این گروه استفاده کمی خواهند داشت.

(۲) طراحی کنترلگر به صورت مستقیم: در این گروه هدف این است که طراحی کنترلگر ربات به صورت مستقیم و با استفاده از ورودی‌ها و خروجی‌های متناظر، صورت گیرد. از جمله پژوهش‌های انجام شده در این گروه می‌توان به [۱۰-۱۴] اشاره کرد؛ که در آن‌ها به ترتیب از شبکه فازی-عصبی، سیستم فازی، یادگیری تقویتی^۶ و ترکیبی از شبکه‌های فازی-عصبی و یادگیری تقویتی در بلوک کنترلگر استفاده شده است.

(۳) استفاده از ماتریس ژاکوبین: فرمان کنترلی اعمالی به ربات، با توجه به قانون کنترل و با استفاده از ماتریس ژاکوبین تولید می‌شود (شکل (۲)).

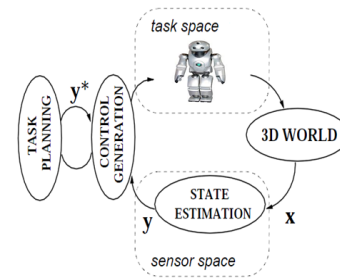


شکل ۲: دیگرام "استفاده از ماتریس ژاکوبین"

(۵) روش ترکیبی: از جمله مقالات این گروه که از ترکیب کنترل خودفرمان سنتی و روشی هوشمند استفاده می‌کنند، می‌توان به [۳] اشاره کرد؛ اما از مشکلات روش ارائه شده در این مقاله، می‌توان به محاسبات زیاد و گسسته‌سازی انجام شده، اشاره کرد.

در این مقاله و طبق مطالب بیان شده، بدلیل دردسترس نبودن مدل ربات و گریز از مشکلات پیشرو برای تقریب آن، همچنین گریز از محاسبات و ساختار پیچیده، از روش‌های موجود در گروه سوم (استفاده از ماتریس ژاکوبین) استفاده خواهیم کرد. ماتریس ژاکوبین ماتریسی است که مؤلفه‌های آن نشان دهنده میزان تغییرات ویژگی‌ها به تغییر در مقادیر متغیرهای مفصل ربات، می‌باشد [۶]، [۷]، [۱۵] و به روش‌های متفاوتی محاسبه می‌شود. ساده‌ترین راه در این گروه این است که ماتریس ژاکوبین را با استفاده از روابط، بدست آورده و در هر تکرار از قوانین کنترلی، متغیرهای z و f در ماتریس ژاکوبین، که به ترتیب بیان کننده عمق فعلی دوربین و مقدار ویژگی‌های فعلی می‌باشند، ثابت و تقریبی از این مقادیر در محل مطلوب دوربین در نظر گرفته شوند

ورودی (u) به ربات می‌دهد و در نهایت حرکت ربات در فضای سه بعدی اتفاق می‌افتد [۶].



شکل ۱: پردازش اصلی در کنترل خودفرمان بینایی [۶]

کنترل خودفرمان بینایی از دیدگاه‌های مختلف از جمله تخمین مد بینایی-موتور^۱، پیکربندی حسگرها، تعداد دوربین‌های کاربردی، طراحی تابع خطا برای تخمین، و محل قرارگیری دوربین با توجه به ربات دسته‌بندی می‌شود [۱، ۶]. کنترل خودفرمان بینایی از دیدگاه تخمین مد بینایی-موتور، به زیرگروه‌هایی تقسیم می‌شود که در این مقاله تمرکز ما بر روی زیرگروه "کنترل خودفرمان بینایی بر مبنای تصویر" می‌باشد. در این زیرگروه تصاویر مستقیماً برای تخمین جابجایی مطلوب ربات استفاده می‌شوند [۶، ۷] و به دلیل پیاده‌سازی ساده‌تر، کارآمدتر می‌باشد [۱]. مبنای عملکرد این روش، کاهش خطای فاصله‌ای تصویری بین مجموعه‌ای از ویژگی‌های تصویری فعلی و مطلوب می‌باشد [۵]. استفاده از الگوریتم‌هایی که به عواملی همچون پارامترهای ذاتی دوربین^۲، پارامترهای بیرونی دوربین^۳ و پارامترهای فیزیکی ربات وابسته هستند باعث می‌شود دقت کالیبراسیون این پارامترها تاثیر قابل توجهی بر خطای کنترل بگذارد. همچنین کالیبراسیون کاری زمان‌بر و خسته‌کننده می‌باشد؛ لذا استفاده از الگوریتم‌های مبتنی بر سیگنال‌های تصویری تنظیم نشده^۴ مطلوب‌تر خواهد بود [۱]، [۸].

با مطالعه بر روی کارهای انجام شده، روش‌های متنوعی در حوزه کنترل خودفرمان بینایی ارائه شده؛ که می‌توان آنها را به چهار گروه کلی تقسیم کرد:

(۱) استفاده از مدل ربات (محیط): کنترل بر اساس مدل ربات انجام می‌شود و از جمله کارهای انجام شده، می‌توان به کنترل پیش‌بینی-مدل^۵ اشاره کرد؛ که در [۹] بیان شده است. در بسیاری از موارد، به دست آوردن

¹ Visual-Motor Model Estimation

² Intrinsic Parameter of Camera

³ Extrinsic Parameter of Camera

⁴ Uncalibrated Visual Signal

⁵ Model-prediction Control

⁶ Reinforcement Learning

در کار قبلی ما [۲۱]، برای کنترل بازوی ربات مدل RoboTEK II ST240 [۲۲] تنها از یک وبکم ساده استفاده شد که به همین دلیل امکان کاهش خطا در بعد عمق وجود ندارد. ساختار کنترل-گر، مبنی بر معکوس ماتریس ژاکوبین می‌باشد که با توجه به ماهیت غیرخطی، معکوس ماتریس ژاکوبین با استفاده از شبکه عصبی تقریب زده شد. ضعف‌های عمده کار قبلی شامل عدم امکان کاهش خطا در بعد عمق بخاطر استفاده از یک وبکم ساده که تنها ویژگی‌ها در دو بعد را در دسترس می‌گذاشت، تعیین ماتریس بهره در ساختار کنترلگر مورد استفاده بصورت ثابت و تجربی، و قابلیت‌های پایین ربات است. در این مقاله با استفاده از یک بازوی ربات صنعتی و دوربین کینکت^۳ که علاوه بر تصویر رنگی از محیط، اطلاعات عمق را می‌دهد؛ راهکارهایی برای فائق آمدن بر چالش‌ها و ضعف‌های کار قبلی ارائه می‌شود.

بطور کلی نوآوری‌های این مقاله به شرح زیر می‌باشد:

- ۱) امکان بکارگیری سیستم ارائه شده در فضای سه بعدی با استفاده از دوربین کینکت
- ۲) ارائه سیستم فازی برای تعیین هوشمند ماتریس بهره در ساختار کنترلگر
- ۳) پیاده سازی روش با استفاده از ربات صنعتی
- ۴) انتخاب ویژگی‌های مناسب جهت حصول به دقت مناسب

ساختار مقاله به شرح زیر است: در بخش دوم مفاهیم پایه بیان می‌شود. روش ارائه شده در بخش سوم می‌آید. بخش چهارم حاوی نتایج حاصل از آزمایشات انجام شده بر روی یک بازوی ربات صنعتی است و بخش پنجم بیان کننده نتیجه‌گیری می‌باشد.

۲- مفاهیم پایه

۲-۱- معرفی دوربین کینکت

به دلیل لزوم کار در محیط سه بعدی و صنعتی، لازم است اطلاعات عمق شی (مختصات در بعد Z) نیز بدست آید. روش‌های متنوعی از جمله استفاده از دو دوربین در حالت استریو [۲۳، ۵] و استفاده از دوربین کینکت می‌باشد. دوربین کینکت دوربینی از نوع RGB-D (Red, Green, Blue and Depth) می‌باشد که خروجی سنسور عمق را به همراه تصویر رنگی ترکیب می‌کند و اطلاعات عمق از شی موردنظر را در اختیار قرار می‌دهد [۲۴، ۲۵]. در مقالات [۲۶-۲۸] از این دوربین برای بدست آوردن ویژگی‌های سه بعدی

[۱۶]. گرچه این روش، ساده است؛ لیکن سرعت همگرایی پایین و خطای بالایی دارد. در [۱۷] از فیلتر کالمن-بوسی^۱ برای تخمین ماتریس ژاکوبین استفاده شده است. ضعف اساسی این کار در گسسته سازی انجام شده، می‌باشد.

در مقاله [۱۸] از دو شبکه عصبی یکی برای استخراج ویژگی و دیگری برای تخمین ماتریس ژاکوبین استفاده شده است؛ اما توضیحی در مورد قانون کنترلی استفاده شده و خروجی‌های هر کدام از شبکه‌ها بیان نشده است. در مقاله [۱۹] ورودی‌های شبکه عصبی تقریب زننده معکوس ژاکوبین، تنها وضعیت مجری نهایی است؛ که در عمل دیده می‌شود که با این وضعیت، شبکه عصبی نمی‌تواند تقریب مناسبی را ارائه دهد. همچنین در مقاله [۲۰] با استفاده از بحث پایداری لیاپانف، روابطی برای تازه سازی ماتریس ژاکوبین محاسبه شده است. شایان ذکر است مقالات [۱۸-۲۰] در یک محیط ساده شبیه سازی شده اند و در آنها مواردی چون نحوه پیدا کردن مفاصل در تصاویر، حذف نویز، لحاظ کردن تأخیر و دینامیک‌های موتورهای محرکها لحاظ نشده است.

در مقاله [۱۵]، ماتریس ژاکوبین با استفاده از پروسه یادگیری و با طی مراحل زیر بدست می‌آید:

- ۱) بردن مجری نهایی به محل مطلوب (x_d)
- ۲) فعال شدن حسگرها و بدست آوردن ویژگی‌های محل مطلوب (f_d)
- ۳) در مرحله آخر برای بدست آوردن ماتریس ژاکوبین، مجری نهایی در k موقعیت مکانی پشت سرهم از پیش تعریف شده اطراف محل مطلوب قرار می‌گیرد (این محل‌ها فاصله‌های یکسانی دارند و فقط با تغییر در r آمین درجه آزادی^۲ رخ می‌دهد) و در هر محل، بردار $f_s(r_i)$ (بردار ویژگی‌ها در محل) مقداردهی می‌شود.

ضرایب ستون مربوطه از ماتریس ژاکوبین (با توجه به اینکه کدام درجه آزادی یا متغیر مفصل ربات تغییر کرده است) را ثابت در نظر می‌گیرد و در هر محل آموزش (r_i)، تمام ضرایب ستون مربوطه محاسبه می‌شوند. در نهایت برای بدست آوردن ضرایب ماتریس ژاکوبین از روش تخمین خطی استفاده می‌شود. توجه شود در مقاله مذکور از ساختار خطی برای بدست آوردن ماتریس ژاکوبین استفاده شده است؛ در حالیکه طبیعت سیستم، غیر خطی می‌باشد و لذا در عمل مشاهده می‌شود که دقت مطلوب به دست نمی‌آید.

¹ Kalman-Bucy filter

² Degree of Freedom (DOF)

³ Kinect

از شی استفاده شده است.

۲-۳- ماتریس ژاکوبین

همانگونه که قبلاً هم بیان شد، ماتریس ژاکوبین ماتریسی است که مؤلفه‌های آن نشان دهنده میزان تغییرات ویژگی‌ها به تغییر در مقادیر متغیرهای مفصل ربات می‌باشد؛ پس می‌توان آن را بصورت زیر نمایش داد [۶، ۷، ۱۵]:

$$J(q) = \left[\frac{\partial f}{\partial q} \right] = \begin{bmatrix} \frac{\partial f_1(q)}{\partial q_1} & \dots & \frac{\partial f_1(q)}{\partial q_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_k(q)}{\partial q_1} & \dots & \frac{\partial f_k(q)}{\partial q_m} \end{bmatrix} \quad (2)$$

در رابطه فوق، J ماتریس ژاکوبین، m نشان دهنده درجات آزادی ربات، q نشان دهنده متغیرهای مفصل ربات، k تعداد ویژگی‌های تصویری و f برداری از ویژگی‌های تصویری می‌باشد.

با توجه به آنچه در [۲، ۳، ۷، ۳۰] بیان شده و با فرض اینکه سرعت زاویه‌ای (${}^c\Omega_e$) و سرعت خطی (cT_e) مجری‌نهایی با

توجه به قاب دوربین به ترتیب ${}^c\Omega_e = [w_x, w_y, w_z]^T$ و

${}^cT_e = [v_x, v_y, v_z]^T$ باشند (x, y, z محورهای

مختصات و w_x و v_x به ترتیب سرعت زاویه‌ای و سرعت خطی مجری‌نهایی در راستای محور x است) و نقطه P نقطه‌ای باشد که به مجری‌نهایی متصل است؛ سرعت نقطه P با توجه به قاب دوربین طبق رابطه (۳) می‌باشد:

$${}^c\dot{P} = {}^c\Omega_e \times {}^cP + {}^cT \quad (3)$$

که می‌توان بصورت سیستم سه معادله‌ای زیر نوشت:

$$\dot{x} = z w_y - y w_z + v_x \quad (4)$$

$$\dot{y} = x w_z - z w_x + v_y \quad (5)$$

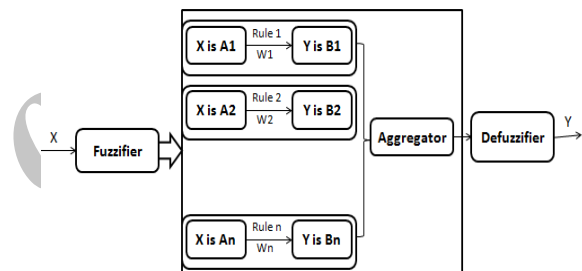
$$\dot{z} = y w_x - x w_y + v_z \quad (6)$$

از آنجا که u و v مختصه‌های تصویر نقطه P در تصویر دوربین می‌باشند، با توجه به شکل (۴) و روابط در دو مثلث مشابه، روابط زیر را داریم:

$$x = \frac{uz}{\lambda}, y = \frac{vz}{\lambda} \quad (7)$$

۲-۲- سیستم فازی

برای سیستم‌های فازی، سیستم‌های مبتنی بر دانش هستند که قابلیت‌های آن‌ها همچون بیان دانش بصورت قواعد اگر-آنگاه، توانایی مدل‌سازی سیستم‌های غیرخطی و نایقین، و بیان دانش به زبان انسانی؛ باعث استفاده روزافزون آن‌ها در سیستم‌های کنترل و هوشمند شده است. شکل زیر نمای کلی یک سیستم استنتاج فازی را نشان می‌دهد. همچنانکه در شکل دیده می‌شود ورودی ابتدا فازی شده و سپس با توجه به قواعد و توابع عضویت تعریف شده در هر بعد، مجموعه فازی منتجه از هر قاعده بدست می‌آید. با استفاده از اپراتور اجتماع این مجموعه‌های منتجه با هم ترکیب شده و مجموعه فازی نهایی حاصل می‌شود. در قدم آخر مقدار خروجی سیستم فازی با استفاده از اپراتور فازی زدا محاسبه می‌شود [۲۹].



شکل ۳: نمای کلی سیستم استنتاج فازی

در این مقاله از سیستم فازی مدل سوگنو مرتبه صفر استفاده شده است که قواعد آن به صورت زیر می‌باشد:

$$R_i: \text{if } x_1 \text{ is } A_{i1} \text{ and } \dots \text{ and } x_n \text{ is } A_{in} \text{ then } y = \bar{y}_i$$

در قاعده فوق x_1 تا x_n ورودی‌های سیستم و A_{i1} تا A_{in} توابع عضویت ورودی می‌باشند. خروجی سیستم نیز با \bar{y} نشان داده شده است. خروجی نهایی سیستم با استفاده از رابطه زیر محاسبه می‌شود:

$$y^* = \sum_{i=1}^c \mu_i \cdot \bar{y}_i \quad (1)$$

در رابطه فوق، c نشان دهنده تعداد قواعد سیستم و μ_i درجه عضویت ورودی به قاعده i ام را نشان می‌دهد.

ویژگی‌ها ذخیره می‌شوند.

۴-۲- فرمان کنترل

با استفاده از روابط و توضیحات ارائه شده در [۶]، [۱۳]، رابطه بین تغییرات ویژگی‌های تصویری (\dot{f}) و تغییرات مختصات مفاصل ربات (\dot{q}) را می‌توان با رابطه (۱۴) بیان کرد. اگر محل تصویری نقاط ویژگی به عنوان معیار سنجش استفاده شود، آنگاه تابع خطا بصورت رابطه (۱۵) تعریف می‌شود:

$$e(f) = f^c - f^* \quad (15)$$

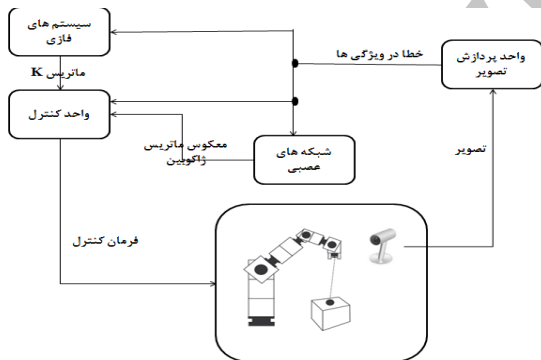
در رابطه فوق f^c نشان دهنده بردار ویژگی در موقعیت فعلی و f^* نشان دهنده بردار ویژگی مطلوب می‌باشد. هدف کنترل، صفر شدن تابع خطای فوق می‌باشد و با این هدف به رابطه زیر خواهیم رسید:

$$u = \dot{q} = K J^\# e(f) \quad (16)$$

که در آن u نشان دهنده بردار فرمان کنترلی، $J^\#$ نشان دهنده معکوس کلی ماتریس ژاکوبین و K ماتریسی قطری با مقادیر ثابت است که ماتریس بهره نامیده می‌شود.

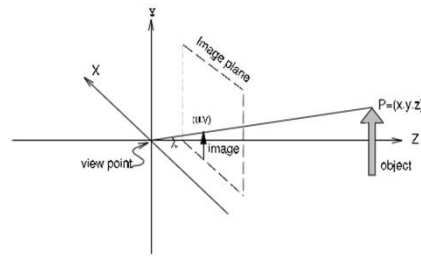
۳- روش ارائه شده

کلیات روش ارائه شده در شکل (۵) نشان داده شده است.



شکل ۵: کلیات روش ارائه شده

در اولین قدم لازم است ویژگی‌های مناسب برای سیستم انتخاب شوند. انتخاب ویژگی‌ها باید با دقت صورت گیرد؛ تا حرکت شی را بصورت یکتا نشان دهد، مقدار آن‌ها در محل مطلوب مشخص باشد [۷] و باعث ایجاد ابهام نشود. از آنجا که لازم است سیستم در فضای سه بعدی بکار رود، فاصله مجری‌نهایی در بعد x ، y و z با هدف (که خروجی بلوک پردازش تصویر می‌باشند) به عنوان ویژگی‌ها در این مسئله، انتخاب شدند. هر یک از بلوک‌های موجود در شکل فوق به شرح زیر می‌باشند:



شکل ۴: دستگاه مختصات دوربین [۳]

با جاگذاری روابط (۷) در روابط (۴) تا (۶) داریم:

$$\dot{x} = z w_y - \frac{vz}{\lambda} w_z + v_x \quad (8)$$

$$\dot{y} = \frac{uz}{\lambda} w_z - z w_x + v_y \quad (9)$$

$$\dot{z} = \frac{z}{\lambda} (v w_x - u w_y) - v_z \quad (10)$$

با استفاده از قاعده مشتق‌گیری از رابطه کسری و روابط (۷) خواهیم داشت:

$$\dot{u} = \lambda \frac{z\dot{x} - x\dot{z}}{z^2} \quad (11)$$

$$\dot{v} = \lambda \frac{z\dot{y} - y\dot{z}}{z^2} \quad (12)$$

با جاگذاری روابط (۱۱) و (۱۲) در روابط فوق خواهیم داشت:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \frac{\lambda}{z} & 0 & \frac{-u}{z} & \frac{-uv}{\lambda} & \frac{\lambda^2 + u^2}{\lambda} & -v \\ 0 & \frac{\lambda}{z} & \frac{-v}{z} & \frac{-\lambda^2 - v^2}{\lambda} & uv & u \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ w_x \\ w_y \\ w_z \end{bmatrix} \quad (13)$$

که ماتریس 2×6 فوق ماتریس ژاکوبین نامیده می‌شود و رابطه (۱۳) را می‌توان بصورت مختصر در رابطه (۱۴) نشان داد:

$$\dot{f} = J\dot{q} \quad (14)$$

در این رابطه، \dot{f} نشان دهنده تغییرات ویژگی‌های تصویری می‌باشد. درنهایت با استفاده از رابطه (۱۴)، می‌توان تخمینی از

سرعت ربات ($\dot{q} = [v_x, v_y, v_z, w_x, w_y, w_z]^T$) بدست آورد؛ تا ویژگی‌ها در صفحه تصویر فعلی به ویژگی‌های مطلوب برسند. به بیان دیگر با این رابطه، حرکت لازم ربات بدست می‌آید؛ تا ربات از محل شروع به محل مطلوب (هدف) برسد. ویژگی‌های مطلوب با روش آموزش با نشان دادن^۱، به ربات نشان داده می‌شوند و این

¹ Teach by Showing

۳-۱- بلوک پردازش تصویر

ورودی این بلوک، تصویر دوربین و خروجی آن، خطای ویژگی‌ها در محل فعلی با هدف می‌باشد. فاصله مجری نهایی در دو بعد X و Y با استفاده از تصویر رنگی خروجی دوربین محاسبه می‌شود. بدین گونه که مجری نهایی را با استفاده از ویژگی رنگ در تصویر مشخص کرده و مختصات آن در دو بعد X و Y بر حسب پیکسل محاسبه می‌شود. سپس مختصات مجری نهایی از مختصات دو بعد X و Y مربوط به هدف کم می‌شود تا فاصله مجری نهایی در دو بعد X و Y بر حسب پیکسل محاسبه شود. فاصله مجری نهایی با هدف در بعد Z ، به روش‌های متنوعی قابل محاسبه می‌باشد و در این مسئله از دوربین کینکت برای محاسبه عمق استفاده شده است. بدین صورت که پس از مشخص شدن محل مجری نهایی، با استفاده از ماتریس عمق خروجی دوربین، عمق مجری نهایی بر حسب میلی‌متر محاسبه و از عمق هدف کم می‌شود؛ تا فاصله در بعد Z بین مجری نهایی و هدف بدست آید.

۳-۲- بلوک شبکه‌های عصبی

طبق رابطه (۲) و مقاله [۲۱]، هریک از المانهای ماتریس ژاکوبین بیان کننده میزان تغییرات ویژگی‌ها به تغییر در مقادیر متغیرهای مفصل ربات می‌باشد و با توجه به ماهیت غیرخطی ماتریس ژاکوبین، از شبکه‌های عصبی برای تقریب معکوس این ماتریس استفاده می‌شود. به منظور جمع‌آوری داده‌های لازم برای آموزش شبکه‌ها، باید سه مفصل بازوی ربات جابجا شوند و نسبت تغییرات در مقادیر مفاصل به تغییرات در ویژگی‌ها محاسبه شود. بنابراین ابتدا دو مفصل پایه و شانه، ثابت در نظر گرفته می‌شود و با تغییرات یک درجه‌ای در مفصل آرنج، میزان تغییر در ویژگی‌های انتخاب شده بدست می‌آید؛ سپس مفصل شانه یک درجه تغییر داده می‌شود و مجدداً مفصل آرنج با فواصل یک درجه‌ای تغییر داده می‌شود. با حرکت مفصل شانه در تمام فضای کاری خود، مفصل پایه یک درجه تغییر داده می‌شود و مجدداً تمام مراحل طی می‌شود. با این روش داده‌های لازم برای آموزش شبکه عصبی و تقریب ماتریس ژاکوبین جمع‌آوری می‌شود. شبه کد مربوط به فاز جمع‌آوری داده، در زیر نشان داده شده است.

```
for i = start1 to end1 with step1
  for j = start2 to end2 with step2
    for k = start3 to end3 with step3
      go to position (i, j, k)
      get picture
      gather needed data
    endfor3
  endfor2
endfor1
```

سپس به ازای هرکدام از درجات آزادی بازوی ربات، یک شبکه عصبی دو لایه با ساختار پیشرو طراحی می‌شود. ورودی‌های هر کدام از شبکه‌ها، مقادیر ویژگی‌ها و حالت سیستم می‌باشد. حالت سیستم با موقعیت سه مفصل بازوی ربات بیان می‌شود اما بخاطر محل قرارگیری دوربین (که تقریباً در راستای خط تقارن پایه و بالای ربات قرار گرفته)، موقعیت مفصل پایه ربات بصورت ضمنی تأثیرگذار خواهد بود و از آنجا که دیگر ورودی‌های شبکه، فاصله مجری نهایی تا هدف (که همان ویژگی انتخاب شده می‌باشد) را نشان می‌دهد؛ پس موقعیت مجری نهایی نیز در شبکه وارد می‌شود. پس برای بیان منحصربه‌فرد حالت سیستم، لازم است موقعیت مفصل شانه ربات، به عنوان ورودی به شبکه داده شود. با توجه به توضیحات ذکر شده برای تأمین ورودی‌های شبکه‌ها، باید محل مفصل شانه و مجری نهایی با استفاده از تصاویر دوربین، مشخص شود؛ لیکن باتوجه به صنعتی و دقیق بودن ربات مورد استفاده، بجای موقعیت مفصل شانه در سه بعد، کافی است زاویه مفصل به شبکه‌ها داده شود. خروجی‌های شبکه‌ها نیز بیان کننده میزان تغییر در مقادیر ویژگی‌ها بر حسب میزان تغییر در متغیر مفصل موردنظر می‌باشند. در نهایت شبکه‌ها با داده‌های جمع‌آوری شده و بصورت برون خط، آموزش داده می‌شوند. ماتریس B در رابطه (۱۷)، معکوس ماتریس ژاکوبین تقریب زده شده، می‌باشد؛ که ابعاد آن n (تعداد درجات آزادی بازوی ربات) در m (تعداد ویژگی‌های انتخاب شده) می‌باشد و هر سطر آن، توسط یکی از شبکه‌های عصبی تقریب زده می‌شود.

$$B = \begin{bmatrix} \frac{\Delta a}{\Delta f_1} & \dots & \frac{\Delta a}{\Delta f_m} \\ \vdots & \ddots & \vdots \\ \frac{\Delta z}{\Delta f_1} & \dots & \frac{\Delta z}{\Delta f_m} \end{bmatrix} \quad (17)$$

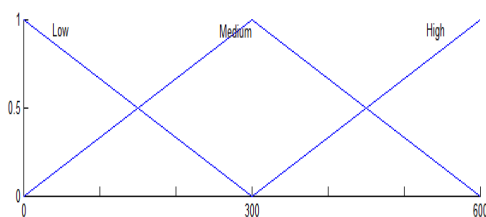
۳-۳- بلوک سیستم‌های فازی

ماتریس بهره K موجود در رابطه (۱۶)، ماتریس قطری با مقادیر ثابت بین ۰ و ۱ می‌باشد. توجه شود که مقادیر این ماتریس نباید خیلی کوچک در نظر گرفته شود؛ چون سرعت رسیدن به هدف (صفر شدن بردار خطا) کم خواهد شد. همچنین نباید خیلی هم بزرگ انتخاب شوند؛ چون بازه حرکت ربات در هر چرخه کنترلی زیاد خواهد شد و اگر خطای اندکی در فرمان اعمالی وجود داشته باشد، تبدیل به جابجایی بزرگی می‌شود. پس امکان دارد تعداد گام‌ها در رسیدن به هدف زیاد شود و یا ربات از هدف عبور کند. در اینجا برای تعیین ماتریس K از سیستم فازی سوگنو مرتبه صفر استفاده می‌شود. دانش این سیستم بر اساس تجربیات بدست آمده



شکل ۶: نمای کلی از سیستم کنترل بازوی ربات

برای تعیین مقادیر ماتریس بهره K در رابطه (۱۸) از سه سیستم فازی سوگنو مرتبه صفر استفاده شد. ورودی این سیستم‌های فازی، قدرمطلق فاصله مجری‌نهایی در بعد x ، y و z با هدف می‌باشد و خروجی هر کدام از سیستم‌ها، درایه ماتریس K می‌باشد. شکل زیر نشان دهنده توابع عضویت ورودی در سیستم فازی شماره ۱ (مربوط به محاسبه درایه k_{11}) می‌باشد.



شکل ۷: توابع عضویت ورودی‌های سیستم فازی شماره ۱

قواعد موجود در سیستم فوق به شرح زیر می‌باشد:

if e_1 is Low then k_{11} = Low.

if e_1 is Medium then k_{11} = Medium.

if e_1 is High then k_{11} = High.

که در قواعد فوق e_1 بیان کننده فاصله مجری‌نهایی در بعد x با هدف می‌باشد. مقدار تالی قواعد (Low، Medium و High) به ترتیب اعداد ۰،۳، ۰،۵ و ۰،۷ می‌باشد.

برای محاسبه معکوس ماتریس ژاکوبین از سه شبکه عصبی، هر کدام برای یکی از متغیرهای مفصل بازوی ربات، استفاده شده است. برای جمع‌آوری داده‌های لازم برای آموزش شبکه‌ها، ابتدا مفصل پایه و شانه را در ابتدایی‌ترین موقعیت خود (زاویه ۳۰۰ درجه برای مفصل پایه و زاویه ۵۱۲ برای مفصل شانه) قرار دادیم. سپس مفصل آرنج را با اختلاف زاویه‌ای ۳۰ درجه‌ای، از ابتدا تا انتهای فضای کاری آن حرکت دادیم. در گام بعد به مفصل شانه ۳۰ درجه افزودیم و مجدداً مفصل آرنج را در تمام فضای کاری حرکت دادیم تا اینکه مفصل شانه نیز کل فضای کاری خود را

بدین صورت است: اگر فاصله مجری‌نهایی تا هدف کم (Low) باشد درایه‌های قطری ماتریس K عدد ۰،۳، اگر فاصله مجری‌نهایی تا هدف متوسط (Medium) باشد، عدد ۰،۵ و اگر فاصله مجری‌نهایی تا هدف زیاد (High) باشد عدد ۰،۷ در نظر گرفته می‌شود. ورودی هریک از سیستم‌های فازی، قدرمطلق فاصله مجری‌نهایی تا هدف در سه بعد x ، y و z می‌باشد و خروجی سیستم‌ها k_{ij} ، $i = 1, 2, 3$ می‌باشد.

۴-۳- بلوک واحد کنترل

بلوک واحد کنترل در شکل (۴) براساس رابطه (۱۶) طراحی می‌شود. بدین ترتیب که خروجی سیستم‌های فازی، خروجی شبکه‌های عصبی و خروجی واحد پردازش تصویر به عنوان ورودی این واحد می‌باشند و خروجی این واحد، میزان تغییر در متغیرهای مفصل بازوی ربات (Δq_i) می‌باشد. رابطه (۱۸) فرمان کنترلی خروجی واحد کنترل را بیان می‌کند.

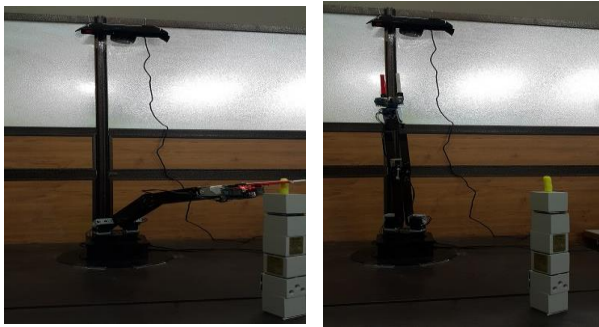
$$u_i = \Delta q_i = S_i(e(F)) * N_i(A) * e(F), i = 1..d \quad (18)$$

در رابطه فوق S_i نشان دهنده سیستم فازی، N_i شبکه عصبی، i شماره متغیر مفصل بازوی ربات و d تعداد درجات آزادی (موتورهای مفصل) ربات می‌باشد.

۴- آزمایشات

در این مقاله آزمایشات بر روی بازوی ربات صنعتی با سه درجه آزادی دورانی (از گروه ربات‌های آرنجی) انجام می‌شود. به منظور جمع‌آوری تصاویر لازم برای واحد پردازش تصویر، از دوربین کینکت که بصورت ثابت و بالای بازوی ربات نصب شده، استفاده شده است. با استفاده از این دوربین و عملیات پردازش تصویری لازم، می‌توان مختصات مجری‌نهایی را در سه بعد محاسبه کرد. سپس با استفاده از سیستم طراحی شده، فرمان‌های کنترلی مناسب را تولید کرد تا خطا در هر سه بعد کاهش یابد و بازوی ربات به هدف تعیین شده برسد. نمای کلی سیستم در شکل (۶) نشان داده شده است. فیلم‌های مربوط به آزمایشات انجام شده را می‌توان در سایت آزمایشگاه هوش محاسباتی و رباتیک دانشگاه یزد [۳۱] مشاهده نمود.

تصویر 640×480 پیکسل می‌باشد.



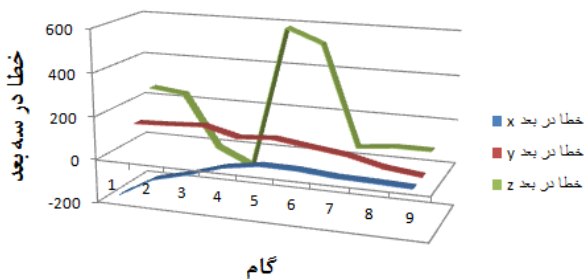
(ب)

شکل ۹ ب: موقعیت نهایی

(الف)

شکل ۹ الف: موقعیت ابتدایی

همانطور که در شکل (۱۰) مشخص است، در ابتدای حرکت، خطای بین مختصات مجری‌نهایی با هدف در سه بعد به ترتیب ۱۷۳، -۱۲۲ و ۲۵۵ می‌باشد. با اعمال اولین فرمان کنترلی به بازوی ربات و حرکت آن به سمت هدف (در اولین گام از پروسه کنترل بازو)، خطای مجری‌نهایی با هدف ۸۳، -۱۲۹ و ۲۳۵ می‌شود. حرکت ربات با توجه به فرمان کنترلی اعمالی ادامه می‌یابد و خطای بین مجری‌نهایی و هدف، به کمتر از ۷ پیکسل کاهش می‌یابد؛ تا مجری‌نهایی به نقطه ۱۵۳، ۴۰۱ و ۱۴۱۳ می‌رسد.



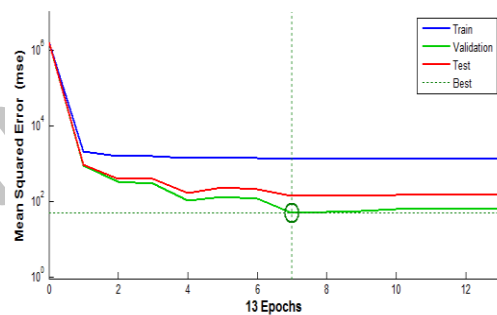
شکل ۱۰: خطای ویزگی‌های تصویری برحسب پیکسل و میلی‌متر

در شکل (۱۱) نمودار مجذور مربعات خطا در آزمایش مذکور نشان داده شده است.

در آزمایشات برای تعیین ماتریس بهره K از سیستم فازی استفاده شد. اگر آزمایش فوق را بدون استفاده از سیستم فازی طراحی شده و با در نظر گرفتن مقدار ثابت ۰٫۵ برای ماتریس K انجام دهیم؛ تعداد گام‌ها برای رسیدن مجری‌نهایی به هدف افزایش خواهد یافت. در شکل (۱۲ الف) سیگنال‌های کنترلی به مفاصل در سیستم با استفاده از ماتریس ثابت K و در شکل (۱۲ ب) عملکرد با استفاده از سیستم فازی طراحی شده نشان داده شده است.

پیمود سپس مفصل پایه را 110° درجه حرکت داده و همین عملیات را مجدداً تکرار نمودیم. با استفاده از داده‌های جمع‌آوری شده، ورودی‌ها و خروجی‌های لازم برای آموزش شبکه‌ها فراهم شدند. هر شبکه، چهار ورودی (فاصله مجری‌نهایی در بعد x ، y و z با هدف و زاویه مفصل شانه بازوی ربات) و سه خروجی (سطر متناظر با مفصل مربوطه در معکوس ماتریس ژاکوبین در رابطه (۱۷)) خواهد داشت. شایان ذکر است واحد فاصله در بعد x و y پیکسل و در بعد z میلی‌متر می‌باشد.

شکل (۸) تغییرات تابع هزینه (میانگین مربعات خطا) در حین آموزش را برای شبکه اول (مربوط به مفصل پایه ربات) نشان می‌دهد. این شبکه عصبی در طی ۱۳ تکرار توانسته است خروجی را با خطای قابل قبولی تقریب بزند. همچنانکه در شکل دیده می‌شود، خطا برای داده تست نیز بسیار پایین ($RMSE = 5$) می‌باشد که حاکی از قابلیت تعمیم شبکه عصبی تنظیم شده می‌باشد.



شکل ۸: تغییرات تابع هزینه در حین آموزش برای شبکه اول

در ادامه از شبکه‌های عصبی آموزش دیده، برای بدست آوردن مقدار فرمان موتورها طبق رابطه (۱۸) استفاده می‌شود.

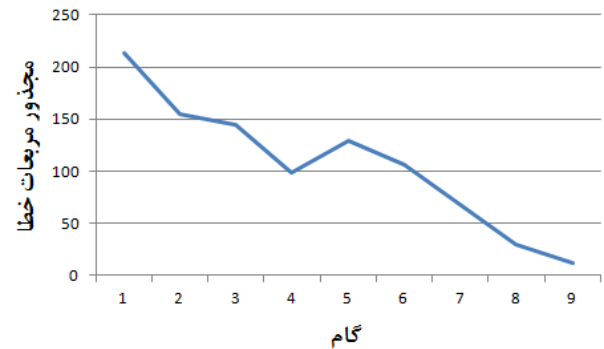
۴-۱- بیان نتایج آزمایشات

مقادیر مفاصل ربات در اولین آزمایش و در ابتدای حرکت 400 ، 530 و 2048 می‌باشد (یا مختصات مجری‌نهایی در قاب دوربین $(333, 285, 1205)$ می‌باشد. قابل ذکر است عدد مربوط به مقدار عمق مجری‌نهایی به میلی‌متر می‌باشد) و هدف در نقطه با مختصات $(160, 407, 1460)$ قرار دارد. در شکل (۹ الف)، محل قرارگیری بازوی ربات در ابتدای عملیات و هدف تعیین شده، مشخص است. پس از ۸ فرمان کنترلی، مجری‌نهایی با دقت خوبی به نقطه نهایی رسیده است (شکل ۹ ب)). در این مثال در نهایت فاصله مجری‌نهایی تا هدف در بعد x ، y و z به ترتیب ۷ پیکسل، ۶ پیکسل و ۴۷ میلی‌متر می‌باشد. قابل ذکر است اندازه ماتریس

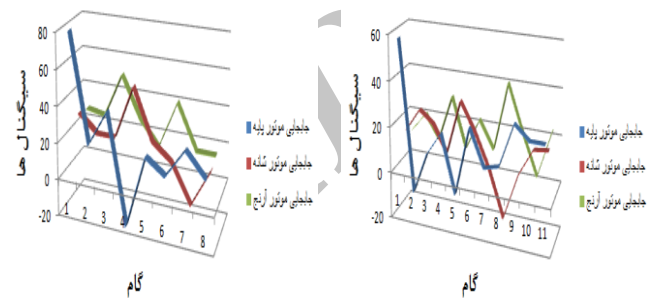
نشان داد که روش ارائه شده می‌تواند مجری نهایی را از هر مکان در فضای سه بعدی به نقطه دلخواه با دقت قابل قبول برساند. همچنین استفاده از سیستم فازی برای تعیین ماتریس بهره K باعث افزایش سرعت رسیدن به هدف به میزان ۳۰ درصد می‌شود. به عنوان کار آینده می‌توان کارایی روش ارائه شده در مسئله تعقیب مسیر را بررسی نمود.

مراجع

- [1] H. Wang, M. Jiang, W. Chen, and Y. Liu, "Visual servoing of robots with uncalibrated robot and camera parameters," *Mechatronics*, vol. 22, no. 6, pp. 661–668, 2012.
- [2] S. Hutchinson and F. Chaumette, "Visual Servo Control," *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp. 82–90, 2006.
- [3] M. Mitic, M. Lazarevic, B. Babic, and Z. Miljkovic, "Neural network Reinforcement Learning for visual control of robot manipulators," *Expert Syst. with Appl.*, vol. 40, no. 5, pp. 1721–1736, 2013.
- [4] H. Wang, Y. Liu, and W. Chen, "Visual tracking of robots in uncalibrated environments," *Mechatronics*, vol. 22, no. 4, pp. 390–397, 2012.
- [5] P. P. Kumar and L. Behera, "Visual servoing of redundant manipulator with Jacobian matrix estimation using self-organizing map," *Rob. Auton. Syst.*, vol. 58, no. 8, pp. 978–990, 2010.
- [6] D. Kragic, H. I. Christensen, and others, "Survey on visual servoing for manipulation," *Comput. Vis. Act. Percept. Lab. Fiskartorpsv*, vol. 15, no. 1, 2002.
- [7] S. Hutchinson, G. D. Hager, and P. Corke, "A Tutorial on Visual Servo Control," *IEEE Trans. Robot. Autom.*, vol. 12, no. 6, pp. 651–670, 1996.
- [8] F. Janabi-Sharifi, L. Deng, and W. J. Wilson, "Comparison of Basic Visual Servoing Methods," *Mechatronics, IEEE/ASME Trans.*, vol. 16, no. 5, pp. 967–983, Oct. 2011.
- [9] G. Allibert and E. Courtial, "Predictive Control for Constrained Image-Based Visual Servoing," *IEEE Trans. Robot.*, vol. 26, no. 5, pp. 933–939, 2010.
- [10] M. Sadeghzadeh, D. Calvert, and H. A. Abdullah, "Self-Learning Visual Servoing of Robot Manipulator Using Explanation-Based Fuzzy Neural Networks and Q-Learning," *J. Intell. Robot. Syst.*, vol. 78, no. 1, pp. 83–104, 2015.
- [11] I. Siradjuddin, L. Behera, T. M. McGinnity, and S. Coleman, "Image Based Visual Servoing of a 7 DOF robot manipulator using a distributed fuzzy proportional controller," in *IEEE International Conference on Fuzzy Systems*, pp. 1–8, 2010.
- [12] I. H. Suh and T. W. Kim, "Fuzzy membership function based neural networks with applications to the visual servoing of robot manipulators," *IEEE Trans. Fuzzy Syst.*, vol. 2, no. 3, pp. 203–220, 1994.
- [13] Y. Takahashi, M. Takeda, and M. Asada, "Continuous Valued Q-learning for Vision-Guided Behavior Acquisition," in *International Conference on Multisense Fusion and Integration for Intelligent Systems*, no. 1, pp. 255–260, 1999.
- [14] M. P. Deisenroth, C. E. Rasmussen, and D. Fox, "Learning to control a low-cost manipulator using data-efficient reinforcement learning," in *International Conference on Robotics, Science & Systems*, pp. 57–64, 2011.
- [15] D. I. Kosmopoulos, "Robust Jacobian matrix estimation for image-based visual servoing," *Robotics and Computer Integrated Manufacturing*, vol. 27, no. 1, pp. 82–87, 2011.
- [16] F. Chaumette and S. Hutchinson, "Visual Servo Control," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.



شکل ۱۱: مجذور مربعات خطا در سه بعد



شکل ۱۲ الف: سیگنال‌های کنترلی به مفاصل با ماتریس ثابت ۰٫۵
شکل ۱۲ ب: سیگنال‌های کنترلی به مفاصل با استفاده از سیستم فازی

با توجه به شکل (۱۲) استفاده از سیستم‌های فازی برای تعیین ماتریس K باعث کاهش ۳۰ درصدی تعداد گام برای رسیدن به هدف شده است.

۵- نتیجه‌گیری

در این مقاله راهکاری جهت کنترل ربات با تصویر گرفته شده از دوربین کینکت، برای رسیدن از هر نقطه شروع دلخواه به هدف تعیین شده در فضای سه بعدی ارائه شد. قانون کنترل در نظر گرفته شده برای این منظور، مبتنی بر ماتریس ژاکوبین می‌باشد و از معکوس ماتریس ژاکوبین برای محاسبه فرمان کنترلی استفاده می‌کند. از شبکه‌های عصبی مصنوعی برای تقریب ماتریس ژاکوبین استفاده شد. فاصله مجری نهایی در بعد x ، y و z با هدف، و زاویه مفصل شانه به عنوان ورودی در نظر گرفته شدند. در قانون کنترلی، برای محاسبه خروجی کنترلگر لازم است ماتریس قطری بهره K تعیین گردد. بدین منظور سه سیستم خبره فازی مدل سوگنو مرتبه صفر برای تعیین مقدار درایه‌های ماتریس K در هر گام از حرکت بازوی ربات ارائه گردید. روش ارائه شده برای کنترل یک بازوی ربات صنعتی مورد استفاده قرار گرفت. نتایج آزمایشات

- [17] J. Qian and J. Su, "Online estimation of image jacobian matrix by kalman-bucy filter for uncalibrated stereo vision feedback," in *IEEE International Conference on Robotics and Automation*, vol. 1, no. 1, pp. 562–567, 2002.
- [18] K. Stanley, Q. M. J. Wu, A. Jerbi, and W. A. Gruver, "Neural network-based vision guided robotics," in *IEEE International Conference on Robotics and Automation*, pp. 281–286, 1999.
- [19] J. Pei, S. X. Yang, and G. S. Mittal, "Vision Based Robot Control Using Position Specific Artificial Neural Network," in *2010 International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 110–115, 2010.
- [20] C. C. Cheah, C. Liu, and J. J. E. Slotine, "Adaptive Jacobian vision based control for robots with uncertain depth information," *Automatica*, vol. 46, no. 7, pp. 1228–1233, 2010.
- [21] F. Nadi, V. Derhami, and M. Rezaeian, "Vision Based Robot Manipulator Control with Neural Modeling of Jacobian Matrix." , vol. 8, no. 2, pp. 11–20, 2014.
- [22] "LJ Group," <http://www.ljgroup.com>. access: june 2014.
- [23] R. Gascon B and M. Barraza R, "Six DOF Stereoscopic Eye-in-Hand Visual Servo System BIBOT," in *Robotics Symposium and Latin American Robotics Symposium (SBR-LARS), 2012 Brazilian*, pp. 284–289, 2012.
- [24] C. Teuliere and E. Marchand, "Direct 3D servoing using dense depth maps," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1741–1746, 2012.
- [25] G. J. Garcia and F. Torres, "Guidance of Robot Arms using Depth Data from RGB-D Camera," in *Proceedings of the 10th International Conference on Informatics in Control, Automation and Robotics*, pp. 315–321, 2013.
- [26] J. Stowers, M. Hayes, and A. Bainbridge-Smith, "Altitude control of a quadrotor helicopter using depth map from Microsoft Kinect sensor," in *IEEE International Conference on Mechatronics (ICM)*, pp. 358–362, 2011.
- [27] I. Siradjuddin, L. Behera, T. M. McGinnity, and S. Coleman, "A position based visual tracking system for a 7 DOF robot manipulator using a Kinect camera," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2012.
- [28] T. Leyvand, C. Meekhof, Y.-C. Wei, J. Sun, and B. Guo, "Kinect Identity: Technology and Experience," *Computer (Long Beach Calif)*, vol. 44, no. 4, pp. 94–96, Apr. 2011.
- [29] L. X. Wang, *A Course in Fuzzy Systems and Control*. Prentice Hall PTR, 1997.
- [30] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley & Sons New York, 2006.
- [31] "Laboratory of Computational Intelligence and Robot," https://pws.yazd.ac.ir/lcir/?page_id=659

Vision Based Robot Manipulator Control in 3D Space Using Fuzzy System and Neural Modeling

Farzane Nadi¹, Vali Derhami^{2*}

1- Computer Engineering Department, Yazd University, Yazd, Iran.

2*- Corresponding Author: Computer Engineering Department, Yazd University, Yazd, Iran.

¹farzane.nadi@yahoo.com, ^{2*}vderhami@yazd.ac.ir

Abstract- Visual servoing system controls a robot by visual feedback so that robot moves from any arbitrary start position to the target positions. The coordinates of points in three dimensions is needed in 3D space. In this paper, a Kinect camera is used to collect RGB images as well as workspace's depth matrix. The control law is obtained using Jacobian matrix. Since, the mathematical model of robot and workspace, is unknown, artificial neural networks is applied to approximate inverse of Jacobian matrix by gathering data. The approximated neural models are used in control law directly. For each degree of freedom of the robot manipulator, a two-layer feedforward neural network is considered. The distance between end-effector and target in 3D space, and the shoulder joint coordinates are inputs of each of the networks and outputs are the fraction of the related robot joint changes to the image features changes (the elements of inverse of Jacobian matrix). The proposed method has been implemented on an industrial robot manipulator. The experimental results show that the proposed control system can move the end-effector to different target positions in workspace with good accuracy and fewer steps in comparison with the previous method.

Keywords- Visual Servoing, Jacobian Matrix, Fuzzy System, Artificial Neural Networks.