

دومین کنگره مشترک سیستمهای فازی و هوشمند ایران

2nd Joint Congress on Fuzzy and Intelligent Systems

طراحی کنترلر فازی Takagi-Sugeno ترکیب شده با روش جای دهی قطب بهینه در فضای حالت برای کنترل ربات موازی 3-RRR

کاوه کمالی، علیرضا اکبرزاده توتونچی

دانشگاه فردوسی مشهد، دانشکده مهندسی، گروه مکانیک

E-mail: kaveh.kamali@gmail.com

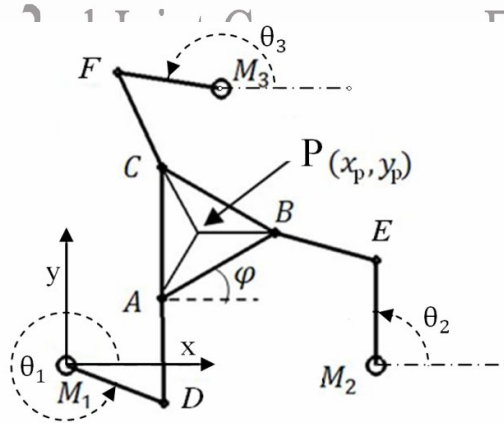
چکیده- ربات‌های موازی در چند دهه اخیر مورد توجه بسیاری از پژوهشگران قرار گرفته‌اند. کنترلرهای ارائه شده برای این نوع ربات‌ها به علت ساختار پیچیده و دینامیک غیر خطی ربات موازی معمولاً کنترلرهای پیچیده‌ای هستند که پایدار نبوده و به علت زمان زیادی که برای محاسبات نیاز دارند در عمل قابل پیاده سازی نیستند. در این مقاله سعی شده تا با ترکیبی از کنترل خطی بهینه و مدل فازی کنترلری سریع و در عین حال با دقت مناسب ارائه شود. ابتدا با استفاده از خطی سازی دینامیک پیچیده ربات موازی حول چندین نقطه‌ی تعادل مختلف در فضای کاری ربات، برای هر نقطه تعادل یک کنترلر بهینه در فضای حالت طراحی شده است. سپس با استفاده از یک سیستم فازی Takagi-Sugeno مقدار ماتریس کنترلر برای کل فضای کاری ربات تخمین زده می‌شود. در نهایت این کنترلر برای یک مسیر سینوسی شکل در فضای Simulink نرم‌افزار Matlab آزمایش شد که نتایج عملکرد مناسب کنترلر طراحی شده را نشان می‌دهد.

کلید واژه- کنترل ربات موازی، سیستم فازی Takagi-Sugeno، کنترلر بهینه.

1- مقدمه

سینماتیک مستقیم آن حل شود. سینماتیک مستقیم ربات- های موازی دارای حل تحلیلی نیست و همچنین بسیار زمان‌بر و دارای خطا است. به این دلیل پژوهشگران برای کنترل ربات‌های موازی به روش‌های که نیاز به مدل دینامیکی ندارند مانند کنترلر فازی، کنترلر بر اساس شبکه عصبی و کنترلرهای تطبیقی روی آوردند [3] [4] [5]. این روش‌ها دارای سرعت بالایی هستند اما دقت کنترلر در آنها پایین می‌آید. در برخی موارد نیز پژوهشگران سعی کردند با ساده‌سازی دینامیک ربات و در نظر نگرفتن برخی عوامل و ترکیب روش بر پایه مدل با روش‌های مقاوم به کنترلر سریع‌تر و دقیق‌تری دست پیدا کنند [6]. این روش‌ها دارای دقت بالاتری هستند و مقاوم نیز می‌باشند اما سرعت محاسبات در آنها هنوز مشکل‌ساز است. در این مقاله ابتدا با استفاده از خطی‌سازی، مدل خطی ربات حول نقطه‌ی تعادل به دست آورده شده است سپس با استفاده از روش کنترلر بهینه خطی در فضای حالت ماتریس کنترلر فیدبک برای هر کدام از این نقاط تعادل محاسبه شد. با توجه به این نکته که با تغییر نقطه تعادل مقدار ماتریس فیدبک نیز تغییر می‌کند برای پیدا کردن مقدار تخمینی ماتریس

ربات‌های موازی به عنوان «یک زنجیره سینماتیکی بسته که در آن نگه دارنده ابزار توسط چندین زنجیره سینماتیکی مجزا به پایه ربات متصل شده است» تعریف می‌شوند [1]. با توجه به سختی بالا، قابلیت کار در سرعت‌های بالا و قابلیت حمل بارهای سنگین در ربات‌های موازی این ربات‌ها از دهه‌ی 1990 در مقالات علمی و همچنین در صنعت بسیار مورد توجه قرار گرفته‌اند [1]. بسیاری از محققان مسئله کنترل ربات‌ها را مورد بررسی قرار داده‌اند. کنترلر ربات‌ها در دو شاخه مختلف دسته‌بندی می‌شود: کنترلر بر اساس مدل و کنترلر بدون استفاده از مدل دینامیکی ربات [2]. در روش کنترلر بر اساس مدل از مدل دینامیکی ربات برای کنترل استفاده می‌شود و در نتیجه پاسخ دقیق‌تر است. از روش‌های کنترلر بر اساس مدل می‌توان به روش گشتاور محاسبه شده اشاره کرد [2]. اما استفاده از مدل دینامیکی ربات مشکلاتی از قبیل مقاوم نبودن و زمان زیاد محاسبات را دارد. مشکل زمان در ربات‌های موازی چندین برابر است. به این دلیل که دینامیک این ربات‌ها بسیار پیچیده است. علاوه بر این در ربات‌های موازی برای استفاده از دینامیک ربات ابتدا باید



شکل 1- ربات موازی صفحه‌ای 3-RRR [7]

3- دینامیک ربات موازی 3-RRR

در این بخش دینامیک ربات موازی 3-RRR به روش مکمل متعامد طبیعی NOC مورد بحث قرار می‌گیرد. ربات 3-RRR دارای $r = 7$ عضو صلب باشد که توسط $m = 9$ اتصال یک درجه آزادی به یکدیگر متصل شده‌اند. تغییر مکان کل سیستم را توسط یک بردار m بعدی که شامل مختصات همه اتصالات می‌باشد، به صورت زیر می‌توان بیان کرد.

$$\mathbf{q} = [q_1 \quad q_2 \quad \dots \quad q_m]^T \quad (1)$$

در این رابطه q_i مختصات اتصال i ام ($i = 1, 2, \dots, m$) می‌باشد. در ربات 3-RRR تمام اتصالات از یکدیگر مستقل نیستند. بنابراین بردار موقعیت اتصالات \mathbf{q} را می‌توان به صورت زیر تعریف کرد:

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}'' \\ \mathbf{q}''' \end{bmatrix} \quad (2)$$

که در آن \mathbf{q}'' بردار n بعدی شامل مختصات اتصال مستقل و \mathbf{q}''' بردار $m-n$ بعدی شامل مختصات اتصالات غیر مستقل می‌باشند. با توجه به اینکه تعداد درجات آزادی در ربات 3-RRR برابر با 3 است برای این ربات داریم $n = 3$. فضای کارترین ربات 3-RRR شامل موقعیت صفحه نگهدارنده‌ی ابزار (مرکز مثلث ABC) و زاویه ابزار است که آن‌ها را با بردار $\mathbf{p} = [x \quad y \quad \varphi]^T$ نشان می‌دهیم.

فیدبک در هر نقطه دلخواه از فضای حالت از مدل فازی Takagi-Sugeno استفاده شده است. سپس از ماتریس فیدبک که توسط مدل فازی به دست آمده است برای کنترل ربات استفاده شده است. برتری این روش این است که علاوه بر اینکه دارای سرعت بسیار بالایی است دقت بالایی نیز دارد. در این مقاله کنترل ربات صفحه‌ای 3-RRR مورد بررسی قرار گرفته است. ابتدا سینماتیک ربات بررسی شده است. سپس دینامیک ربات با استفاده از روش تکمیل کننده‌ی متعامد طبیعی (NOC) به دست آمده است. در مرحله بعد ماتریس فیدبک حالت بهینه حول چندین نقطه تعادل محاسبه شده است و از این ماتریس‌ها برای آموزش

مدل فازی Takagi-Sugeno استفاده شده است. سپس ماتریس فیدبک بدست آمده از مدل فازی برای کنترل ربات به کار رفته است. در نهایت این کنترلر برای یک مسیر سینوسی شکل در فضای Simulink نرم‌افزار Matlab آزمایش شد.

2- ساختار ربات موازی صفحه‌ای 3-RRR

در این مقاله کنترل ربات موازی صفحه‌ای 3-RRR مورد بررسی قرار گرفته است. مدل ربات در شکل (1) مشاهده می‌شود. این ربات شامل سه زنجیره از لینکها (M_1DA ، M_2EB و M_3FC) می‌باشد که صفحه‌ی ابزار ربات را به پایه‌ی ربات متصل می‌کند. سه موتور بر روی پایه‌ی ربات روی رأسهای یک مثلث متساوی الاضلاع در نقاط M_1 ، M_2 و M_3 ثابت شده‌اند و صفحه ابزار ربات مثلث متساوی‌الضلاع ABC است. هر زنجیره سینماتیکی شامل دو لینک یک مفصل چرخشی ساده می‌باشد. ساختار کلی ربات به صورت سه زنجیره حلقه بسته M_1DABEM_2 ، M_2EBCFM_3 و M_3FCADM_1 سینماتیکی است که دو حلقه از آنها مستقل هستند و یکی وابسته است. هدف این ربات این است که ابزار را در موقعیت و زاویه دلخواه در صفحه قرار دهد.

$$(J_x)_{ij} = \frac{\partial f_i}{\partial x_j} \quad i, j = 1 \dots 6 \quad (7)$$

$$(J_u)_{ij} = \frac{\partial f_i}{\partial u_j} \quad i, j = 1 \dots 6 \quad (8)$$

$$\mathbf{I}(q)\ddot{\mathbf{q}}^a + \mathbf{C}(q, \dot{\mathbf{q}})\dot{\mathbf{q}}^a - \mathbf{G}(q) = \boldsymbol{\tau}^a \quad (3)$$

پس از خطی سازی بر اساس بسط سری تیلور، با صرف نظر کردن از ترمهای بالاتر داریم:

$$\Delta \dot{\mathbf{x}}(t) = \mathbf{A}(t)\Delta \mathbf{x}(t) + \mathbf{B}(t)\Delta \mathbf{u}(t) \quad (9)$$

که در آن $\mathbf{A}(t) = J_x(\mathbf{x}_0, \mathbf{u}_0, t)$ و $\mathbf{B}(t) = J_u(\mathbf{x}_0, \mathbf{u}_0, t)$

5- محاسبه‌ی ماتریس فیدبک حالت بهینه

$$\boldsymbol{\tau}^a = [\tau_1^a \quad \tau_2^a \quad \dots \quad \tau_n^a] \quad (4)$$

برای بهینه سازی سیستم خطی شده‌ی معادله‌ی (9) از معیار انتگرال درجه دوم استفاده می‌کنیم. با استفاده از این روش ماتریس بهره بهینه به صورت زیر به دست می‌آید [9]:

$$\mathbf{K} = \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} \quad (10)$$

که ماتریس \mathbf{P} باید معادله‌ی زیر را برآورده سازد:

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} - \mathbf{P} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} + \mathbf{Q} = 0 \quad (11)$$

که در آن \mathbf{Q} یک ماتریس متقارن غیر منفی است. کمیت $\mathbf{x}^T(t) \mathbf{Q} \mathbf{x}(t)$ مقیاسی از میزان انحراف حالت در زمان t از حالت صفر است. همچنین \mathbf{R} ماتریس وزنی متقارن معین مثبت است. جمله‌ی $\mathbf{u}^T(t) \mathbf{R} \mathbf{u}(t)$ نیز مقیاسی از دامنه‌های ورودی (گشتاور اعمالی به ربات) است. در این پژوهش برای حل معادله (11) از جعبه ابزار کنترل مقاوم (Robust Control Toolbox) در نرم افزار MATLAB استفاده شده است.

6- مدل فازی برای تخمین ماتریس کنترل بهینه

در هر نقطه از فضای کاری

همان گونه که بیان شد می‌توان با خطی سازی ربات حول نقاط تعادل، آن را با استفاده از یک ماتریس کنترل فیدبک به صورت بهینه کنترل کرد. اما به علت اینکه دینامیک ربات موازی غیرخطی است، حول هر نقطه‌ی تعادل ماتریس فیدبک متفاوتی به دست می‌آید. در اینجا برای اینکه بتوان با سرعت بالا مقدار ماتریس فیدبک را حول هر نقطه‌ی دلخواه تخمین زد، از مدل فازی Takagi-Sugeno استفاده شده است. همان طور که در خطی سازی مشاهده شد، متغیرهای فضای حالت $(q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3)$ هستند. بنابراین در

تکمیلی طبیعی (NOC)

۷ الی ۹ آبان ماه ۱۳۸۷

Ma و Angeles [8] مدل دینامیکی ربات موازی را به صورت زیر به دست آوردند:

که در آن ماتریس‌های $\mathbf{I}(q) \in \mathbb{R}^{3 \times 3}$ ماتریس اینرسی،

$\mathbf{C}(q, \dot{q}) \in \mathbb{R}^{3 \times 3}$ ماتریس مربوط به نیروهای کوریولیس و

$\mathbf{G}(q) \in \mathbb{R}^{3 \times 1}$ ماتریس نیروهای گرانشی هستند که توسط

Ma و Angeles [8] تعریف شده‌اند. بردار گشتاورهای وارد

بر مفاصل محرک نیز به صورت زیر تعریف می‌شود:

4- خطی سازی دینامیک ربات

همان گونه که مشاهده شد دینامیک ربات موازی کاملاً غیر-خطی است و برای اینکه بتوانیم از روش‌های معمول کنترل خطی برای کنترل آن استفاده کنیم باید مدل خطی شده ربات را حول یک نقطه تعادل به دست آوریم. برای این کار از یک روش کاربردی و موفق در خطی سازی بر اساس بسط سری تیلور (Taylor series) استفاده شده است [9]. ابتدا مدل دینامیک ربات را به فضای حالت می‌بریم. با تعریف متغیرهای $x_1 = q_1, x_2 = q_2, x_3 = q_3, x_4 = \dot{q}_1, x_5 = \dot{q}_2, x_6 = \dot{q}_3$ به عنوان متغیرهای حالت، معادلات حالت دینامیک ربات به صورت زیر می‌شود:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_6 \end{bmatrix} = f(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_6(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \\ \dots \end{bmatrix} = f'(\mathbf{q}, \dot{\mathbf{q}}) \quad (5)$$

که در آن داریم:

$$f'(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{I}(\mathbf{q})^{-1} (\boldsymbol{\tau}^a - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}^a + \mathbf{G}(\mathbf{q})) \quad (6)$$

ماتریس $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$ را ماتریس جاکوبین (Jacobian) تابع f

نسبت به \mathbf{x} نامیده و آن را به صورت J_x نشان می‌دهیم.

همچنین ماتریس $\frac{\partial f(\mathbf{u})}{\partial \mathbf{u}}$ را ماتریس جاکوبین تابع f

نسبت به \mathbf{u} نامیده و آن را به صورت J_u نشان می‌دهیم

هر نقطه‌ای تعادل مقادیر همه‌ی این متغیرها باید مشخص باشند. اما با توجه به اینکه با بالا رفتن تعداد متغیرها تعداد قوانین فازی به صورت تصاعدی بالا می‌رود در این مقاله برای سادگی از مقادیر سرعت‌های زاویه‌ای $(\dot{q}_1, \dot{q}_2, \dot{q}_3)$ صرف نظر شده است و در همه‌ی نقاط تعادل مقدار آنها برابر صفر در نظر گرفته شده است.

$$f_{ij}(\mathbf{x}) = \exp\left(\frac{(x_j - \mu_{ij})^2}{2\sigma_{ij}^2}\right) \quad (14)$$

در بخش بعد نحوه به دست آوردن پارامترهای μ_{ij} و σ_{ij} شرح داده خواهد شد.

7- آموزش مدل فازی سینماتیک مستقیم

برای آموزش مدل فازی از نمونه داده‌ها استفاده شده است. همان طور که بیان شد قوانین فازی متناظر با هر کدام از داده‌های ورودی و خروجی ایجاد شده است. برای تولید قوانین مراحل زیر به کار رفته است:

1. فضای کاری ربات را مشخص می‌کنیم و نقاطی را با پراکندگی یکنواخت از درون فضای کاری ربات در نظر می‌گیریم.

2. برای هر یک از این نقاط با استفاده از سینماتیک معکوس زوایای موتورهای محرک (q_1, q_2, q_3) مربوط به آن‌ها را به دست می‌آوریم.

3. مقدار ماتریس کنترل بهینه $K(q^a, \dot{q}^a)$ هر کدام از دسته زوایای q^a به دست آمده از مرحله‌ی 4 محاسبه می‌کنیم. مقادیر \dot{q}_i را برای سادگی صفر در نظر می‌گیریم.

4. زوایای به دست آمده برای مفصل‌های محرک (q^a) و سرعت زاویه‌ای مفصل‌های محرک (\dot{q}^a) به عنوان ورودی و ماتریس $K(q^a, \dot{q}^a)$ به عنوان خروجی مدل فازی در نظر گرفته می‌شود.

5. پارامترهای μ_{ij} و σ_{ij} را به صورت زیر تعیین می‌کنیم:

$$\mu_{ij} = \mathbf{x}_i(j) \quad (15)$$

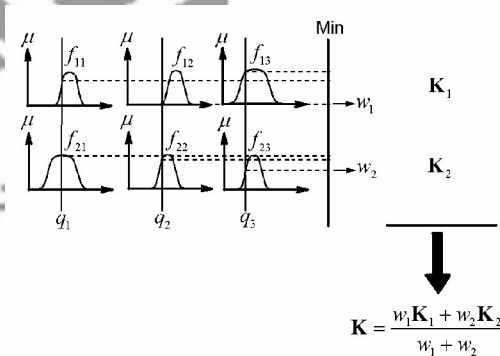
$$\sigma_{ij} = \max(|\mathbf{x}_i(j) - \mathbf{x}_{i-1}(j)|, |\mathbf{x}_{i+1}(j) - \mathbf{x}_i(j)|) \quad (16)$$

که در آن $\mathbf{x}_i(j)$ نشان دهنده درایه‌ی i ام از نقطه‌ی تعادل i ام است.

مدل فازی Sogeno (که همچنین به عنوان مدل FSK نیز شناخته می‌شود) توسط Takagi, Sogeno, Kang [10] در تلاش برای ایجاد کردن قوانین فازی به صورتی سینماتیک از یک سری نمونه داده‌های ورودی و خروجی معرفی شد. در این پژوهش از مدل فازی Sugeno مرتبه صفر برای تخمین ماتریس فیدبک $K(q^a, \dot{q}^a)$ برای کنترل ربات استفاده شده است. نمودار مدل فازی Sugeno در شکل (2) مشاهده می‌شود.

پس از اعمال غیرفازی سازی خروجی مدل فازی Takagi-Sugeno که مقدار تخمین زده شده برای ماتریس K است، به صورت معادله زیر به دست می‌آید:

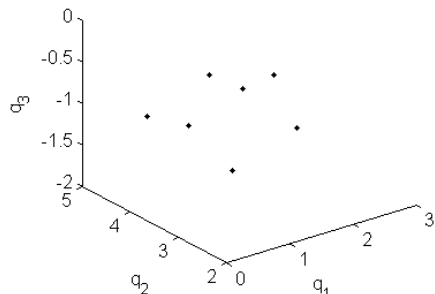
$$\mathbf{K} = \frac{\sum_{i=1}^r w_i(\mathbf{x}) \mathbf{K}_i}{\sum_{i=1}^r w_i(\mathbf{x})} \quad (12)$$



شکل 2- مدل فازی Takagi-Sugeno

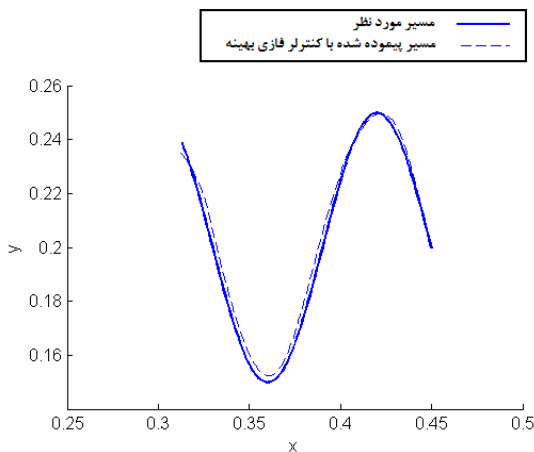
که در آن \mathbf{K}_i ماتریس بهره بهینه به دست آمده برای نقطه-ی i ام است و $w_i(\mathbf{x})$ به صورت زیر به دست می‌آید:

$$w_i(\mathbf{x}) = \prod_{j=1}^n f_{ij}(\mathbf{x}) \quad (13)$$



شکل 3- نقطه‌های تعادل نشان داده شده در فضای زوایای مفصل‌های محرک

پس از محاسبه شدن ماتریس‌های بهره برای هر نقطه از مقادیر زوایای موتورهای مربوط به هر نقطه به عنوان ورودی و از مقدار ماتریس K به دست آمده برای هر نقطه به عنوان خروجی برای هر نمونه آموزش سیستم فازی استفاده می‌شود. همان‌طور که گفته شد، یک مسیر سینوسی برای آزمایش تخمینگر فازی در نظر گرفته شده است در نظر گرفته شده است. مدل فازی در نرم‌افزار MATLAB شبیه سازی شد. نمودار مسیر طی شده برای کنترلر ارائه شده در این مقاله در شکل (4) نشان داده شده است. خطای اندکی که در تعقیب مسیر توسط این کنترلر مشاهده می‌شود نشان دهنده‌ی کارا بودن این کنترلر است. این در حالی است که زمان محاسبات در این کنترلر بسیار کمتر از کنترلرهای بر پایه مدل می‌باشد.



شکل 4- نتیجه به دست آمده توسط کنترلر فازی بهینه

9- نتیجه گیری

در این مقاله کنترلر ربات صفحه‌ای 3-RRR مورد بررسی

برای بررسی نتایج به دست آمده از کنترلر ارائه شده در این مقاله، یک مسیر خاص در فضای کاری ربات در نظر گرفته شده است. برای سادگی فرض شده است که زاویه‌ی ابزار در طول مسیر ثابت و معادل صفر باشد. بنابراین فضای کار ربات از سه بعد (x, y, φ) به دو بعد (x, y) کاهش می‌یابد. مشخصات ربات که برای شبیه سازی در نظر گرفته شده است در جدول (1) نشان داده شده است. برای آموزش ربات 7 نقطه به صورت یکنواخت در صفحه xy ربات پراکنده شده‌اند در نظر گرفته شده است. سپس توسط برنامه نوشته شده برای 3-RRR در نرم‌افزار MATLAB زوایای محرک‌های معادل این 7 نقطه به دست آمد که در شکل (3) این زوایا به صورت نقاطی در فضای زوایای محرک‌ها نشان داده شده است.

جدول 1- مشخصات ربات موازی 3-RRR شبیه‌سازی شده

	زنجیره		
	اول	دوم	سوم
l_1 (m)	0/25	0/25	0/15
l_2 (m)	0/25	0/25	0/15
l_3 (m)	0/25	0/25	0/15
موقعیت مفصل محرک (m)	$\begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$	$\begin{Bmatrix} 0.7 \\ 0 \end{Bmatrix}$	$\begin{Bmatrix} 0.35 \\ 0.7\frac{\sqrt{3}}{2} \end{Bmatrix}$

هرکدام از این نقاط به عنوان یک نقطه تعادل در نظر گرفته می‌شود و ماتریس بهره‌ی بهینه برای آن نقطه محاسبه می‌شود. مقادیر ماتریس‌های Q و R در جدول (2) نشان داده شده است.

جدول 2- مقادیر ماتریس‌های Q و R برای کنترلر بهینه ربات

نام ماتریس	مقدار در نظر گرفته شده برای ماتریس
Q	$\begin{bmatrix} 100 \times \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & 10 \times \mathbf{I}_{3 \times 3} \end{bmatrix}$
R	$10 \times \mathbf{I}_{3 \times 3}$

دومین کنگره مشترک سیستمهای فازی و هوشمند ایران 2nd Joint Congress on Fuzzy and Intelligent Systems

28-30 October 2008

قرار گرفت ابتدا دینامیک ربات با استفاده از روش تکمیل کننده‌ی متعامد طبیعی (NOC) به دست آمد. در مرحله بعد ماتریس فیدبک حالت بهینه حول چندین نقطه تعادل محاسبه شده است و از این ماتریسها برای آموزش مدل فازی Takagi-Sugeno استفاده شده است. سپس ماتریس فیدبک بدست آمده از مدل فازی برای کنترل ربات به کار گرفته شد. در نهایت این کنترلر برای یک مسیر همیونی شکل در فضای Simulink نرم افزار Matlab آزمایش شد. نتایج نشان دهنده‌ی کارا بودن این کنترلر است. این در حالی است که زمان محاسبات در این کنترلر بسیار کمتر از کنترلرهای بر پایه مدل می باشد.



مراجع

انجمن سیستمهای فازی ایران

- [1] Merlet J.P. and 1988, "Parallel manipulators", in: Proceedings of Seventh CISMIFTO MM Symposium on Theory and Practice of Robots and Manipulators, Udine, Italy, pp. 317-324.
- [2] He, J.F., Jiang, H.Z., Cong, D.C., Ye, Z.M. and Han, J.W., 2007, "A Survey On Control Of Parallel Manipulator", Key Engineering Materials vol. 339, pp. 307-313.
- [3] Su, Y., Sun, D., Ren, L., and Mills, J. K., 2006, "Integration of Saturated PI Synchronous Control and PD Feedback for Control of Parallel Manipulators", IEEE Transactions on Robotics, vol. 22, pp. 202-207.
- [4] Honegger M., Codourey A. and Burdet E., "Adaptive control of the Hexaglide, a 6 dof parallel manipulator", Robotics and Automation, 1997. Proceedings., 1997 IEEE.
- [5] Aly A., Ohuchi H., "Fuzzy hybrid control for positioning a six-degree-of-freedom parallel manipulator", Yamanashi district conference, 2002.
- [6] Lee S.H., Song J.B., Choi W.C., Hong D., "Position control of a Stewart platform using inverse dynamics control with approximate dynamics", Mechatronics. Vol.13(2003), p. 605-619. "
- [7] Gosselin C. and Angeles J., 1989, Kinematics of parallel manipulators, McGill University Montreal, Quebec, Canada, December.
- [8] Ma, O. and Angeles, J., 1989, "Direct Kinematics and Dynamics of a Planar 3-DOF Parallel manipulator", Advances in Design Automation, Proc. Of ASME Design and Automation Conference, 3, pp. 313-320.
- [9] خاکی صدیق ع., اصول کنترل مدرن, انتشارات دانشگاه تهران, 1382.

Intelligent Systems
Scientific Society Of Iran