دومین کنگره مشترک سیستمهای فازی و هوشمند ایران
2nd Joint Congress on Fuzzy and Intelligent Systems
28-30 October 2008 ۷ الی ۹ آبان ماه ۱۳۸۷

تحلیل یادگیری تقویتی در فرایندهای مارکوف به صورت سیستم‌های دیجیتال

سیدمصطفی کلامی هریس
دانشجوی کارشناسی ارشد مهندسی کنترل
دانشگاه فردوسی مشهد
sm.kalami@gmail.com

ناصر پریز
استادیار دانشکده‌ی مهندسی
دانشگاه فردوسی مشهد
n-pariz@ferdowsi.um.ac.ir

محمدباقر نقیبی سیستانی
استادیار دانشکده‌ی مهندسی
دانشگاه فردوسی مشهد
mb-naghibi@ferdowsi.um.ac.ir

چکیده – فرایند تصمیم‌گیری مارکوف یا MDP، یکی از مسائلی است که دارای کاربردهای وسیعی در زمینه‌های مختلف علمی، مهندسی، اقتصادی و مدیریت است. بسیاری از فرایندهای تصمیم‌گیری دارای خاصیت مارکوف می‌باشند و به صورت یک مسأله‌ی تصمیم‌گیری مارکوف قابل بیان هستند. یادگیری تقویتی یکی از مسائلی است که برای حل MDP به کار می‌رود، که به نوبه‌ی خود از برنامه‌ریزی پویا یا DP استفاده می‌کند. در این مقاله معادله‌ی بازگشتی مورد استفاده در بحث یادگیری تقویتی و DP برای حل MDP، به صورت یک معادله‌ی دینامیکی یک سیستم دیجیتال یا گسسته-زمان بازنویسی شده است. به این ترتیب این امکان به وجود آمده است که بتوان با بهره‌گیری از روش‌های موجود در کنترل دیجیتال، به بررسی خواص معادلات به دست آمده پرداخت و تحلیل مناسبی از رفتار عامل یادگیرنده، تحت سیاست‌های مختلف، به عمل آورد. به عنوان مثال، روش مذکور برای تحلیل یک مسأله‌ی جدولی استفاده شده است. نتایج به دست آمده، نشان می‌دهند که یک سیاست بهینه در چارچوب کنترل دیجیتال، به صورت سیستم مرده نَوش قابل توصیف است.

کلید واژه- برنامه‌ریزی پویا، سیستم‌های کنترل دیجیتال، فرایندهای تصمیم‌گیری مارکوف، کنترل تصادفی، یادگیری تقویتی.

۱- مقدمه

از دیدگاه هوش مصنوعی، هدف از یادگیری، به دست آوردن قابلیت تصمیم‌گیری برای حل مسائل مختلف است. این توانایی هنگامی محقق می‌شود که، نگاشتی خاص از فضای حالات به فضای اعمال پیدا شود، که از جهاتی بهتر از سایر نگاشت‌های ممکن باشد [1,2]. یادگیری هنگامی رخ می‌دهد که عامل یادگیرنده (یا به اختصار، عامل) در اثر تجاربی که کسب می‌کند، به نحوی متفاوت و احتمالا بهتر، عمل کند. به این ترتیب می‌توان گفت: یادگیری، تاثیری است که عامل از محیط اطراف می‌گیرد. تاثیری که محیط بر روی عامل دارد، غالبا به صورت یک پسخورد یا فیدبکِ حاوی اطلاعات است که تابعی از حالت عامل و همچنین عملی است که وی انجام می‌دهد [1-4].

اگر عاملی موظف باشد که صرفا با تکیه برپاسخ‌هایی که از محیط دریافت می‌کند، نحوه‌ی عملکرد بهینه را یاد بگیرد، در واقع از عامل خواسته شده است که یک مسأله‌ی *یادگیری تقویتی*[1] را حل کند. پسخوردهایی که محیط به عامل برمی‌گرداند، اطلاعاتی در مورد خوبی یا بدی کارهایی که عامل انجام می‌دهد، در بر دارند و عامل موظف است با استفاده از این پسخوردها، یاد بگیرد که چه عملی در چه شرایطی خوب و در چه شرایطی بد است. در یادگیری تقویتی، پاسخ‌های محیط به صورت اسکالر و با نام پاداش تعریف می‌شوند و عامل وظیفه دارد، در یک بازه‌ی زمانی، مجموع پاداش‌های دریافتی را بیشینه کند. یادگیری تقویتی در جانوران و البته انسان، سهم قابل توجهی از یادگیری را به خود اختصاص می دهد. هنگامی که کسی دست خود را در معرض حرارت قرار می‌دهد و دستش می‌سوزد، در اثر ناشی از این کار، یاد می‌گیرد که دوباره این کار را انجام ندهد. لذت‌ها و دردها، درواقع پاسخ‌هایی هستند که از طرف محیط به موجودات زنده داده می‌شوند، و تعیین کننده‌ی الگوی رفتاری بسیاری از موجودات می‌باشند [1-4].

یکی از انواع مسائلی که به وفور در کاربردهای علمی و

[1] Reinforcement Learning

مهندسی ظاهر می‌شود، مسأله‌ی *فرایند تصمیم‌گیری مارکوف*[2] یا MDP می‌باشد [5-7]. این مسأله را می‌توان در قالب یک مسأله‌ی یادگیری تقویتی بیان نمود و یا استفاده از روش‌هایی که در یادگیری تقویتی مطرح هستند، به حل آن پرداخت [1-3]. یکی از روش‌های مهم در حل MDP، روش *برنامه‌ریزی پویا*[3] یا DP است. این روش که توسط بلمَن [8] در سال ۱۹۵۷ معرفی شده است، یکی از روش‌های قدرتمند برای حل انواع مسائل بهینه‌سازی است. اما کاربردی که این روش در حل MDP ها دارد، در زمینه‌ی یادگیری تقویتی، بیشتر مورد توجه بوده است. ارتباطی که امروزه بین یادگیری تقویتی و روش DP وجود دارد، محصول تحقیق و مطالعه‌ای است که از سال ۱۹۶۱ توسط مینسکی شروع شده است و تا امروز ادامه دارد [1]. در این میان نگارش‌های جدیدی از روش DP با اعمال تغییرات بر روی این روش و یا ترکیب آن با سایر زمینه‌ها، به وجود آمده‌اند. *برنامه‌ریزی پویای تقریبی*[4]، *برنامه‌ریزی پویای افزایشی*[5] و *برنامه‌ریزی پویای عصبی*[6] از جمله نگارش‌های تغییر یافته‌ی DP می‌باشند [9,10].

در این نوشتار فرایند حل مسائل MDP با استفاده از روش DP، به صورت یک دینامیک گسسته-زمان و یا سیستم دیجیتال بیان شده است. سیستم دیجیتالی که به دست می‌آید با روش حلی که برای مسأله ارائه شده است، متناظر است و می‌توان با تحلیل خصوصیات کنترلی این سیستم، خواص روش حل متناظر با آن را مشخص نمود و کیفیت جواب نهایی را حدس زد. استفاده از این معادل‌سازی، این امکان را به وجود می‌آورد که بتوان به تحلیل عملکرد یادگیری تقویتی در محیط‌های مارکوف پرداخت. سایر بخش‌های اسن مقاله به صورت زیر می‌باشند. در بخش ۲، یادگیری تقویتی در محیط‌های مارکوف و روش DP به اختصار توضیح داده می‌شوند. در بخش ۳، روش DP به صورت یک سیستم دیجیتال بیان می‌شود. بخش ۴ نیز، حاوی حل و بررسی یک مسأله‌ی نمونه با استفاده از مطالب مذکور در بخش‌های قبلی می‌باشد.

۲- یادگیری تقویتی در محیط‌های مارکوف

بخش اعظمی از کارهای تحقیقاتی انجام شده بر روی یادگیری تقویتی، توأم با این فرض بوده‌اند که، تعامل بین عامل و محیط اطرافش را می‌توان به صورت یک *فرایند تصمیم‌گیری مارکوف* یا MDP گسسته در زمان مدل‌سازی کرد. مهم‌ترین اجزای تشکیل دهنده‌ی یک فرایند مارکوف محدود و گسسته عبارتند از [2,5-7]:

- یک زمان‌سنج سراسری به صورت $t = 0,1,\ldots,T$ برای شمارش زمان گسسته. (T می‌تواند نامحدود باشد.)
- فضای حالت S که مجموعه‌ی محدودی از حالت‌ها به صورت $\{s^1,s^2,\ldots,s^n\}$ است.
- مجموعه‌ی محدودی از اعمال به صورت $A_s = \{a^{1,s},a^{2,s},\ldots,a^{m,s}\}$ که در هر حالت $s \in S$ قابل انتخاب هستند. اجتماع مجموعه‌ی اعمال ممکن، در تمام حالات ممکن، مجموعه‌ی A را می‌سازد.
- یک تابع تغییر حالت به صورت $P^a_{ss'}$ که احتمال تغییر حالت از s به s' را، در اثر انتخاب عمل a در حالت s مشخص می‌کند.

برای استفاده از یادگیری تقویتی در چنین محیطی، جزء دیگری نیز به این صورت افزوده می‌شود:

- تابع پاداش که به ازای هر سه‌تایی s، a و s'، یک کمیت تصادفی با توزیع ثابت را ایجاد می‌کند. امید ریاضی این کمیت تصادفی $R^a_{ss'}$ است.

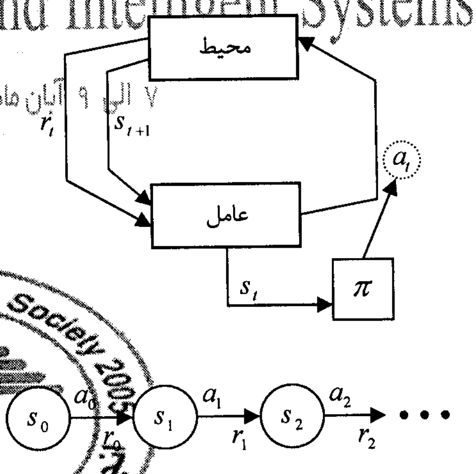
فرض کنید عامل در زمان یا مرحله‌ی t در حالت s_t قرار دارد و عمل $a_t \in A_{s_t} \subseteq A$ را انجام می‌دهد. عامل با احتمال $P^{a_t}_{s_t s'}$ در زمان $t+1$ به حالت s' می‌رود و خواهیم داشت: $s_{t+1} = s'$. ضمنا عامل در نتیجه‌ی این عمل، پاداشی اسکالر به اندازه‌ی r_t خواهد گرفت که در حالت کلی، کمیتی تصادفی و با امید ریاضی $R^{a_t}_{s_t s_{t+1}}$ می‌باشد.

علاوه بر خواص فوق، اصلی‌ترین خاصیت هر فرایند مارکوف به این شکل قابل بیان است: در هر زمان t، احتمال رسیدن به حالت s_{t+1}، صرفا به s_t و a_t وابسته است و مستقل از حالات و اعمال قبل از زمان t می‌باشد. به عبارت دیگر می‌توان نوشت [2,5-7]:

$$\Pr\left(s_{t+1}\middle|s_t,a_t,s_{t-1},a_{t-1},\ldots\right) = \Pr\left(s_{t+1}\middle|s_t,a_t\right) \quad (۱)$$

---

[2] Markov Decision Process
[3] Dynamic Programming
[4] Approximate Dynamic Programming
[5] Incremental Dynamic Programming
[6] Neural Dynamic Programming



شکل ۱ – نحوه‌ی ارتباط عامل با محیط اطرافش

در مورد یادگیری تقویتی، فرض بر این است که تابع پاداش نیز، دارای خاصیت مارکوف است [2]. یعنی:

$$\Pr\left(r_t \mid s_t, a_t, s_{t-1}, a_{t-1}, \ldots\right) = \Pr\left(r_t \mid s_t, a_t\right) \qquad (۲)$$

احتمال انتخاب عمل a از طرف عامل، هنگامی که در حالت s قرار دارد، با نگاشتی به صورت $\pi : \mathrm{S} \times \mathrm{A} \rightarrow [0,1]$ تعریف می‌شود و می‌توان نوشت:

$$\Pr\left(a_t = a \mid s_t = s\right) = \pi(s,a) \qquad (۳)$$

نگاشت π معمولا با نام *سیاست*[7] شناخته می‌شود و مجهول اصلی یک مسأله‌ی یادگیری تقویتی و هر مسأله‌ی تصمیم‌گیری می‌باشد [1-3,5]. نحوه‌ی عملکرد یک عامل با محیط اطرافش، در شکل ۱ به وضوح نشان داده شده است.

برای مقایسه‌ی سیاست‌های مختلف با یکدیگر، می‌توان معیاری را برای سنجش آن‌ها تعریف نمود. این معیار، مقداری است که سیاست در هر حالت از فرایند برمی‌گرداند و به عنوان *خروجی*[8] سیاست در حالت مذکور از آن یاد می‌شود. خروجی یک سیاست، میزانی از پاداش که در اثر اتخاذ تصمیمات متوالی با تبعیت از آن سیاست به دست آمده است. برای هر کدام از حالت‌ها در MDP، ارزشی در نظر گرفته می‌شود. ارزش هر حالت، برابر با امید ریاضی خروجی که با شروع کردن از همان حالت و تبعیت از یک سیاست خاص است. در حالت کلی، منظور از حل یک مسأله‌ی یادگیری تقویتی، پیدا کردن سیاست π^* است به نحوی که مقدار خروجی سیاست و یا ارزش هر کدام از حالت‌ها، بیشینه شوند [1-3]. روش‌های متفاوتی برای تعریف خروجی وجود دارند. روش در اکثر کاربردها معمول است و در این مقاله نیز مورد توجه است، تعریف خروجی به صورت *تنزیلی*[9] می‌باشد. اگر ضریب تنزیل به صورت $\gamma \in [0,1]$ باشد، خروجی تنزیلی به صورت زیر خواهد بود:

$$z_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \qquad (۴)$$

در این حالت، ارزش حالت s به صورت زیر قابل بیان است:

$$\begin{aligned} V^{\pi}(s) &= \mathrm{E}_{\pi}\left(z_t \mid s_t = s\right) \\ &= \sum_{a \in \mathrm{A}} \pi(s,a) \sum_{s' \in \mathrm{S}} \mathrm{P}_{ss'}^{a} \left(\mathrm{R}_{ss'}^{a} + \gamma V^{\pi}(s')\right) \end{aligned} \qquad (۵)$$

که در آن، منظور از E، عملگر امید ریاضی است. اندیس‌های π نیز، صرفا برای تاکید بر این که عامل از سیاست π پیروی می‌کند، نوشته شده‌اند. رابطه‌ی (۵)، به معادله‌ی (بهینگی) بلمن معروف است [1-3,5-10]. یکی از روش‌هایی که برای حل این معادله و یافتن مقدار ارزش تمام حالات استفاده می‌شود، بازگشتی کردن این معادله است. این روش که مبتنی بر *قضیه‌ی نقطه‌ی ثابت*[10] [11] است، پیشنهاد می‌کند که معادله‌ی (۵) به صورت زیر بازنویسی شود:

$$V_{k+1}^{\pi}(s) = \sum_{a \in \mathrm{A}} \pi(s,a) \sum_{s' \in \mathrm{S}} \mathrm{P}_{ss'}^{a} \left(\mathrm{R}_{ss'}^{a} + \gamma V_k^{\pi}(s')\right) \qquad (۶)$$

که در آن $V_k^{\pi}(s)$، تخمین kاُم از مقدار واقعی $V^{\pi}(s)$ است. با توجه به این که $|\gamma| < 1$ است، می‌توان استدلال کرد که $V_{k+1}^{\pi}(s)$ با یک *نگاشت انقباضی*[11] [11] به $V_k^{\pi}(s)$ مرتبط است. طبق قضیه‌ی نقطه‌ی ثابت [11]، این نگاشت دارای نقطه‌ی ثابتِ منحصر به فردی است که جواب معادله‌ی (۵) نیز می‌باشد. با توجه به معادله‌ی (۶)، جواب معادله‌ی (۵) به صورت زیر خواهد بود:

$$V^{\pi}(s) = \lim_{k \to \infty} V_k^{\pi}(s) \qquad (۷)$$

فرایند محاسبه‌ی $V^{\pi}(s)$ برای تمام حالت‌ها، در بحث یادگیری تقویتی به نام *ارزیابی سیاست*[12] [1-3,5] معروف است.

[7] Policy
[8] Return
[9] Discounted
[10] Fixed Point Theorem
[11] Contraction Map
[12] Policy Evaluation

۳- بیان DP به صورت سیستم دیجیتال

فرض کنید برداری به صورت

$$v^{\pi} = \left[V^{\pi}(s^1)\ \ V^{\pi}(s^2)\ \ \cdots\ \ V^{\pi}(s^n)\right]^T \qquad (۸)$$

تعریف شده باشد. این بردار حاوی ارزش تمام حالات یک مدل است. در این صورت می‌توان رابطه‌ی بازگشتی (۶) را به شکل زیر برای تمام حالت بازنویسی کرد و آن را به صورت یک معادله تبدیل نمود:

$$v_{k+1}^{\pi} = \gamma P v_k^{\pi} + R = \gamma \begin{bmatrix} P_{s^1s^1} & \cdots & P_{s^1s^n} \\ \vdots & \ddots & \vdots \\ P_{s^ns^1} & \cdots & P_{s^ns^n} \end{bmatrix} v_k^{\pi} + \begin{bmatrix} R_{s^1} \\ \vdots \\ R_{s^n} \end{bmatrix} \qquad (۹)$$

که در آن داریه‌های ماتریس‌های P و R عبارتند از:

$$P_{ss'} = \sum_{a \in A} \pi(s,a) P_{ss'}^{a} \qquad (۱۰)$$

و

$$R_s = \sum_{a \in A} \sum_{s' \in S} \pi(s,a) P_{ss'}^{a} R_{ss'}^{a} = \sum_{a \in A} \pi(s,a) R_s^{a} \qquad (۱۱)$$

می‌توان معادله‌ی (۹) را به صورت زیر بازنویسی کرد:

$$v_{k+1}^{\pi} = \gamma P v_k^{\pi} + R u_k \qquad (۱۲)$$

که در آن فرض شده است که u_k به ازای تمام مقادیر $k \geq 0$، برابر با واحد باشد، که همان تعریف تابع پله‌ی واحد [12] می‌باشد. معادله‌ی (۱۲) نشان دهنده‌ی یک سیستم گسسته-زمان یا دیجیتال [12] می‌باشد که متغیرهای حالت آن، ارزش‌های مربوط به حالات مختلف می‌باشند. ماتریس حالت این سیستم، از ترکیب اطلاعات مربوط به محیط در قالب $P_{ss'}^{a}$، اطلاعات مربوط به سیاست در قالب $\pi(s,a)$، و ضریب تنزیل به دست آمده است. بردار وروی این سیستم R می‌باشد که اطلاعات مربوط به محیط، سیاست و پاداش‌ها را در بر دارد. طبق قرارداد، ورودی این سیستم، همواره برابر با پله‌ی واحد در نظر گرفته می‌شود. شرط پایداری سیستم فوق، عبارت است از این که، همه‌ی مقادیر ویژه‌ی ماتریس γP، که قطب‌های سیستم (۹) هستند، در داخل دایره‌ی واحد قرار بگیرند [12]. برای تحقق این شرط، می‌بایست ضریب تنزیل γ در نامساوی ذیل صدق کند:

$$\gamma < \frac{1}{\max\limits_{1 \leq i \leq n} |\lambda_i(P)|} = \frac{1}{\rho(P)} \qquad (۱۳)$$

که در آن، $\lambda_i(P)$ نشان دهنده‌ی مقدار ویژه‌ی iاُم، و $\rho(P)$ نیز *شعاع طیفی*[۱۳] ماتریس P می‌باشد. لذا مشاهده می‌شود که شرط $\gamma \leq 1$، الزاما تضمین کننده‌ی همگرایی سری تعریف شده در معادله‌ی (۴) و یا وجود جواب محدود برای (۵) نمی‌باشد. رابطه‌ی (۱۳)، شرط دقیق‌تری برای γ بیان می‌کند. اگر تمام مقادیر ویژه‌ی ماتریس P، داخل دایره‌ی واحد باشند، آن‌گاه سری (۴)، به ازای برخی از مقادیر γ نیز که بزرگتر از یک هستند، همگرا خواهد بود.

۴- حل و بررسی یک مسأله‌ی نمونه

جدولی به صورت شکل نشان داده شده در شکل ۲ را در نظر بگیرید. عاملی (مثلا یک روبات) در یکی از خانه‌های این جدول قرار دارد. این عامل می‌بایست با حرکت در یکی از چهار جهتِ بالا، پایین، چپ و راست، خود را به یکی از دو خانه‌ی هدف، که با رنگ خاکستری مشخص شده‌اند، برساند. حرکت در هر جهت، پاداشی به اندازه‌ی ۱- در پی دارد، که این پاداش منفی، نشان دهنده‌ی هزینه‌ای است که عامل برای هر حرکت می‌پردازد. حرکت‌هایی که باعث خارج شدن عامل از صفحه می‌شوند، بر موقعیت عامل تاثیری ندارند و محل قرار گرفتن عامل در جدول را تغییر نمی‌دهند. عامل بایستی یاد بگیرد که با دریافت بیشترین پاداش و یا پرداخت کمترین جریمه، خود را به یکی از خانه‌های هدف برساند. اگر چنین کاری محقق شود، عامل توانسته است با کمترین تعداد حرکت، به هدف برسد. این مسأله به نام Grid-world شناخته می‌شود و دارای خواص مربوط به فرایند تصمیم‌گیری مارکوف می‌باشد [3-1]. لذا می‌توان برای حل این مسأله، از روش DP استفاده نمود.

G	s^1	s^2	s^3
s^4	s^5	s^6	s^7
s^8	s^9	s^{10}	s^{11}
s^{12}	s^{13}	s^{14}	G

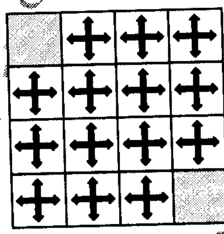
شکل ۲ – شبکه‌ی مربوط به مسأله‌ی مورد بررسی

[13] Spectral Radius

$k=0$

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

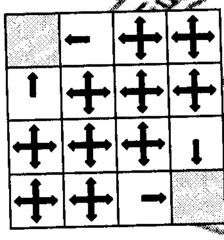
π_0



$k=1$

0	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	0

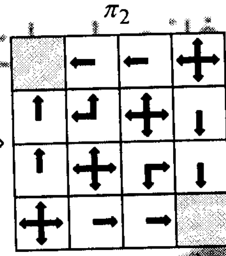
π_1



$k=2$

0	-1.75	-2	-2
-1.75	-2	-2	-2
-2	-2	-2	-1.75
-2	-2	-1.75	0

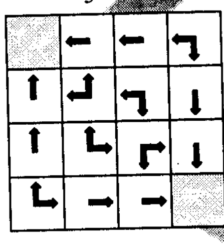
π_2



$k=3$

0	-2.44	-2.94	-3
-2.44	-2.88	-3	-2.94
-2.94	-3	-2.88	-2.44
-3	-2.94	-2.44	0

$\pi_3 = \pi_\infty$



شکل ۳ – چند مرحله از برنامه‌ریزی پویا به ازای سیاست تصادفی

برای استفاده از روش DP، ارزش اولیه‌ی هر کدام از خانه‌ها، برابر با صفر در نظر گرفته می‌شود [1-3,5]. طبق قضیه‌ی نقطه‌ی ثابت، نتیجه‌ی نهایی، مستقل از ارزش اولیه‌ی خانه‌ها می‌باشد و الگوریتم همواره به یک نقطه در فضای جستجو، همگرا می‌شود [11]. سیاستی که تاپایان حل مسأله مورد استفاده قرار گرفته است، سیاست تصادفی است. به این معنی که، در تمام خانه‌های جدول، احتمال حرکت به تمام جهات مساوی، و همگی برابر با چهارم یا ۰/۲۵ می‌باشد. در شکل ۳، چند مرحله از حل تکراری معادله‌ی بلمن، توسط معادله‌ی (۶)، نشان داده شده است. با استفاده از نتایج مربوط به هر مرحله، می‌توان سیاستی را پایه‌ریزی کرد. به این ترتیب که، عامل می‌بایست در هر خانه‌ی جدول، به سمت خانه‌هایی حرکت کند که بیشترین ارزش را دارند. سیاستی که با استفاده از ارزش‌های به دست آمده در مرحله‌ی kاُم به دست می‌آید، به صورت π_k نشان داده شده است. π_0 همان سیاست تصادفی است. π_∞ نیز سیاستی است که با استفاده از ارزش‌های نهایی به دست آمده است و π_∞ و تمام سیاست‌های بعد از آن، همگی معادل هستند.

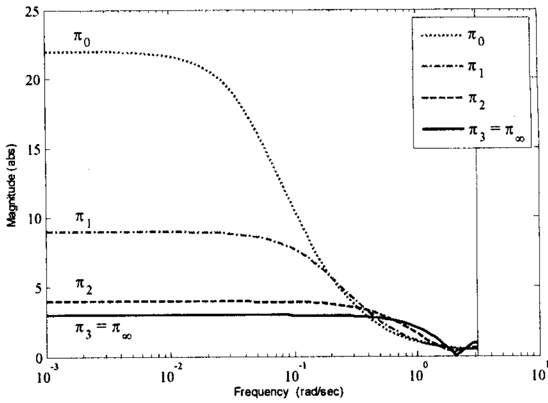
فرض کنید با استفاده از هر کدام از سیاست‌های به دست آمده، معادله‌ی سیستم معرفی شده در معادله‌ی (۹) محاسبه شوند، و ماتریس P در معادله‌ی (۹) برای سیاست π_i به صورت P_i باشد. استفاده از اندیس i، صرفا به دلیل جلوگیری از تداخل اندیس‌ها در معادله‌ی (۹) می‌باشد. شعاع طیفی هر کدام از ماتریس‌های مذکور محاسبه شده‌اند و عبارتند از:

$$\begin{aligned} &\rho(P_0) \cong 0.9468, \quad \rho(P_1) \cong 0.8431, \\ &\rho(P_2) = 0.5, \quad \rho(P_3) = \rho(P_\infty) \cong 0 \end{aligned} \tag{۱۴}$$

به وضوح دیده می‌شود که هر چه قدر سیاست به کار رفته در ایجاد مدل، بهینه‌تر باشد، اندازه‌ی بزرگترین مقدار ویژه‌ی ماتریس حالت نیز کوچک‌تر می‌شود. به خصوص به ازای سیاست π_3 یا همان π_∞، تمامی مقادیر ویژه برابر با صفر است. به این ترتیب، حد بالای ضریب تنزیل، γ، برای همگرایی سری (۴)، به ازای سیاست‌های π_0 تا π_3 به ترتیب عبارت است از: ۱/۰۵۶۲، ۱/۱۸۶۱، ۲ و ∞. به عبارت دیگر، سری تعریف شده با معادله‌ی (۴)، به شرط پیروی از سیاست π_∞، به ازای تمام مقادیر γ همگرا خواهد بود.

از طرفی، قطب‌های سیستم (۹)، با مقادیر ویژه‌ی ماتریس γP برابر هستند. مشاهده می‌شود که قطب‌های سیستم معادل با سیاست π_∞، همگی در مبدا قرار دارند. چنین سیستمی در بحث کنترل دیجیتال، به نام مرده نَوش[14] شناخته می‌شود [12]. یک سیستم مرده نَوش، سریع‌ترین پاسخ ممکن را در بین سیستم‌های هم‌مرتبه‌اش دارد [12]. اگر درجه‌ی چنین سیستمی برابر با n باشد، دقیقا n واحد زمانی (گسسته) طول می‌کشد که پاسخ پله‌ی سیستم به مقدار نهایی برسد و نشست کند. هدف از حل مسأله‌ی فوق نیز، رسیدن به یکی از خانه‌های هدف در کمترین تعداد حرکت می‌باشد. لذا کاملا طبیعی است که پاسخ بهینه، متناظر با یک سیستم مرده نَوش باشد.

[14] Dead Beat



شکل ۴ – پاسخ فرکانسی سیستم‌های تبعیت کننده از سیاست‌های π_0 تا π_∞ در خانه‌ی s^3 از جدول نشان داده شده در شکل ۲

همان طور که در شکل ۳ مشاهده می‌شود، عملکرد سیاست‌های قبل از π_∞، در مورد حالت‌های s^3، s^6، s^9 و s^{12}، مانند π_∞ نمی‌باشد و احتمالا این سیاست‌ها نتایج ضعیفی را برای این حالات در پی دارند. اگر ارزش‌های تمام حالات، به صورت خروجی‌های سیستم دینامیکی (۹) تعریف شوند، این سیستم، یک سیستم یک-ورودی و چند خروجی خواهد بود. تابع تبدیل متناظر با این سیستم، یک بردار با ۱۴ مولفه است. پاسخ فرکانسی این سیستم، حاوی اطلاعات مهمی در مورد محیط و سیاست اعمال بهکار رفته از طرف عامل، می‌باشد. به عنوان مثال، پاسخ فرکانسی مولفه‌ی سوم تابع تبدیل سیستم به ازای سیاست‌های π_0 تا π_∞، در شکل ۴ ترسیم شده است. عملکرد حالت ماندگار برای این مولفه از سیستم، متناظر با مقدار پاسخ فرکانسی در فرکانس صفر است. دیده می‌شود که دامنه‌ی پاسخ فرکانسی در فرکانس صفر، به ازای سیاست‌های π_0 تا π_∞، به ترتیب عبارت است از: ۲۲، ۹، ۴ و ۳. این مقادیر نشان دهنده‌ی متوسط تعداد حرکت‌هایی هستند که با تبعیت از هرسیاست، عامل از s^3 به یکی از خانه‌های هدف می‌رسد. مشاهده می‌شود که به ازای π_∞ کمترین تعداد ممکن از حرکت‌ها به دست آمده است. از روی جدول نشان داده شده در شکل ۲، به وضوح معلوم است که کمترین تعداد حرکت برای رسیدن از s^3 به هر کدام از اهداف، برابر با ۳ حرکت می‌باشد. همچنین در شکل ۴، مشاهده می‌شود که سیستم‌های متناظر با سیاست‌های بهتر، پهنای باند بیشتری دارند. با توجه به این که پهنای باند، معیاری از سرعت پاسخ‌دهی هر سیستم می‌باشد، می‌توان استدلال کرد که برای سیاست‌های بهتر، سرعت پاسخ‌دهی بیشتر می‌باشد.

۵- نتیجه‌گیری

در این مقاله با ارائه‌ی مروری بر یادگیری تقویتی، فرایندهای مارکوف و برنامه‌ریزی پویا، معادلات مربوط به حل یک فرایند مارکوف با استفاده از برنامه‌ریزی پویا، به صورت یک دینامیک گسسته-زمان جمع‌بندی و دوباره‌نویسی شدند. این روش برخورد، امکان بررسی فرایند حل یک مسأله‌ی یادگیری تقویتی در محیط مارکوف را، در قالب یک سیستم دیجیتال فراهم می‌آورد. به این ترتیب، می‌توان شیوه‌های مرسوم در کنترل دیجیتال را برای تحلیل یک فرایند یادگیری استفاده نمود. نتایج حاکی از آن هستند که یک سیاست بهینه، در قالب کنترل دیجیتال به صورت یک سیستم مرده نَوِش قابل توصیف می‌باشد. تعریف یک دوگانی بین فضای تصمیم‌گیری و فضای سیستم‌های کنترل دیجیتال، از مطالعات و تحقیقات تکمیلی است که می‌توان در ادامه‌ی این مقاله، متصور شد.

مراجع

[1] Richard S. Sutton and Andrew G. Barto, *Reinforcement Learing: An Introduction*. MIT Press, Cambridge, MA, 2002.
[2] S. I. Reynolds, "Reinforcement Learning with Exploration," *Ph.D. Thesis*, School of Computer Science, The University of Birmingham, UK, 2002.
[3] Leslie Pack Kaelbling, Michael L. Littman and Andrew W. Moore, "Reinforcement Learning: A Survey," in *Journal of Artificial Intelligence Research*, Vol. 4, pp. 237-285, 1996.
[4] Tom M. Mitchell, *Machine Learning*. McGraw Hill, 1997.
[5] Martin L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley & Sons, Inc., 2005.
[6] Qiying Hu and Wuyi Yue, *Markov Decision Processes with Their Applications*, Springer Science+Busines Media, LLC, 2008.
[7] Heyeong Soo Chang et al., *Simulation-based Algorithms for Markov Decission Processes*, Springer-verlag, London, 2007.
[8] R. E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, 1957.
[9] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, Belmont, 1995.
[10] D. P. Bertsekas and J. N. Tsitsiklis, *Neural Dynamic Programming*, Athena Scientific, Belmont, 1996.
[11] H. Royden, *Real Analysis (3rd Edition)*, Prentice Hall, 1988.
[12] K. Ogata, *Discrete-Time Control Systems (2nd Edition)*, Prentice Hall, 1994.