مجله کنترل
ISSN 2008-8345
جلد ۶، شماره ۳، پاییز ۱۳۹۱، صفحه ۴۹-۳۷

حل عددی معادلات دیفرانسیلی- جبری به شکل هزنبرگ با استفاده از کنترل مد لغزشی

علی خالقی کارسالاری[۱]، مسعود شفیعی[۲]

[۱] فارغ‌التحصیل کارشناسی ارشد مهندسی برق، گروه کنترل، دانشگاه صنعتی امیرکبیر، alikhaleghi@aut.ac.ir

[۲] استاد، دانشکدهٔ مهندسی برق، گروه کنترل، دانشگاه صنعتی امیرکبیر، mshafiee@aut.ac.ir

(تاریخ دریافت مقاله ۱۳۹۱/۵/۲۴، تاریخ پذیرش مقاله ۱۳۹۱/۸/۲۰)

چکیده: در این مقاله روشی برای حل عددی معادلات دیفرانسیلی- جبری به شکل هزنبرگ ارائه شده است. در روش ارائه شده یک سطح لغزشی متناسب با ایندکس سیستم تعریف شده است که معادله‌ی کاملی برای محاسبه‌ی متغیر جبری در اختیار قرار می‌دهد. همچنین به دلیل پایداری سطح لغزشی، همگرایی خطای دوری از خمینه‌ی قید در معادله‌ی دیفرانسیلی- جبری تضمین شده است. در انتها، روش روی چند مثال خطی ایندکس و غیر خطی اعمال شده و نتایج آورده شده است.

کلمات کلیدی: معادلات دیفرانسیلی- جبری، شکل هزنبرگ، حل عددی، کنترل مد لغزشی، ردیابی خمینه‌ی قید.

Solution of Differential-Algebraic Equations in Hessenberg Form Using Sliding Mode Control

Ali Khaleghi Karsalari, Masoud Shafiee

Abstract: In this paper, a method for numerical solution of differential-algebraic equations (DAEs) in Hessenberg form is presented. In this method, a sliding surface proportional to systems index is defined that generates a complete equation to calculate algebraic variables. Since the sliding surface is stable, convergence of the distance from manifold constraint in DAE is satisfied. Finally, the proposed method is applied for some linear and nonlinear index-3 systems. Numerical solutions confirm the accuracy of the proposed technique.

Keywords: differential-algebraic equations, Hessenberg form, sliding mode control, constraint manifold tracking.

۱- مقدمه

رویکردهای مختلفی برای به دست آوردن تقریب عددی پاسخ معادله‌ی دیفرانسیلی- جبری وجود دارد. از جمله‌ی روش‌های ارائه شده، می‌توان به استفاده از فرمول تفاضل وارون [۲,۱]، رانگ-کوتای ضمنی [۲-۴]، تقریب پاده[1] [۵]، تجزیه‌ی آدومین[2] [۶] روش اختلال هوموتپی[3] [۷] و تقریب‌های چند مربعی[4] [۸] اشاره کرد.

به دلیل تنوع بسیار زیاد روش‌های حل عددی موجود، دسته‌بندی آنها اندکی دشوار است. یک دسته‌بندی می‌تواند بر اساس نحوه‌ی ساخت پاسخ باشد. دو رویکرد اصلی، حل ترتیبی و ساخت سراسری پاسخ است.

[1] Pade Approximation
[2] Adomian decomposition
[3] Homotopy Perturbation Method (HPM)
[4] Multiquardatic approximation

[Downloaded from joc.kntu.ac.ir on 2026-11-09]
[DOR: 20.1001.1.20088345.1391.6.3.4.0]

در رویکرد ساخت سراسری پاسخ، که در برخی متون روش‌های طیفی[1] نیز نامیده می‌شوند، سعی می‌شود با استفاده از یک مجموعه از توابع پایه، پاسخ در بازه‌ی حل، تقریب زده شود. لذا هدف به دست آوردن ضرایب توابع پایه به صورتی است که خطا به حد مطلوب برسد. در مقالات از توابع پایه مختلفی بهره برده شده است. از جمله، در [۹] روشی ارائه شده است که از یک چندجمله‌ای درجه‌ی دلخواه S برای تقریب پاسخ استفاده می‌کند. از چندجمله‌ای‌ها لژاندر[2] و چبیشف[3] نیز به عنوان چندجمله‌ای‌های متعامد در تقریب عددی مسائل بسیاری استفاده شده است [۱۰].

در رویکرد ترتیبی، با توجه به معین بودن شرایط اولیه، با استفاده از یک روش گسسته‌سازی، پاسخ برای گام بعدی به دست می‌آید. به طور کلی روش‌های مبتنی بر فرمول تفاضل وارون یا رانگ-کوتای ضمنی جزء این دسته هستند.

در این مقاله روشی با رویکرد ترتیبی برای حل معادلات دیفرانسیلی- جبری که به شکل هزنبرگ داده شده باشند، ارائه شده است. این روش به روش‌های حل معادلات دیفرانسیلی معمولی همراه با قید بسیار نزدیک است. در معادلات دیفرانسیلی معمولی با قید، معادله‌ی قید، اطلاعات اضافه‌ای محسوب می‌شود که بدون آن نیز معادله‌ی دیفرانسیلی معمولی قابل حل است و اگر هیچ خطایی در حل عددی وجود نداشته باشد، معادله‌ی قید باید در پاسخ، برقرار باشد. اما به دلیل وجود خطاهای مختلف در حل عددی، ممکن است این اتفاق صورت نگیرد. لذا از معادله‌ی قید برای اصلاح پاسخ عددی در هر گام استفاده می‌شود. برای این کار سعی می‌شود معادله‌ی پایداری برای همگرا ماندن حل عددی به معادله‌ی قید تشکیل گردد و با استفاده از آن معادله، در هر گام مقدار حالت‌ها اصلاح شود. در روشی که در این مقاله بدان خواهیم پرداخت رویکرد مشابهی استفاده شده است. اما برخلاف معادلات دیفرانسیلی معمولی همراه قید، در معادلات دیفرانسیلی- جبری، قید اطلاعات اضافه محسوب نمی‌شود و بدون آن معادله قابل حل نیست. در روش ارائه شده در این مقاله، معادله‌ی قید به عنوان خطا در نظر گرفته می‌شود و با تشکیل یک معادله‌ی دیفرانسیلی پایدار روی آن، که متناسب با ایندکس سیستم تشکیل می‌شود، حالت‌های جبری دستگاه دیفرانسیلی – جبری محاسبه می‌شود.

ادامه این مقاله به شرح زیر است: در بخش دوم معادلات دیفرانسیلی- جبری تعریف شده‌اند. در بخش سوم روش حل عدد معادلات دیفرانسیلی- جبری شکل هزنبرگ ارائه شده است. در بخش چهارم در مورد انتخاب ضرایب مناسب مربوط به روش ارائه شده توضیحاتی آورده شده است. مثال‌های عددی حل شده با روش ارائه شده و مقایسه با نرم‌افزارهای موجود نیز در بخش پنجم ارائه شده‌اند.

۲- معادلات دیفرانسیلی- جبری

کلی‌ترین شکل معادله‌ی دیفرانسیلی - جبری به صورت زیر است:

$$\begin{aligned}&\mathbf{f}(t,\mathbf{x}(t),\dot{\mathbf{x}}(t))=\mathbf{0},\\&t\in[0,T],\ \mathbf{x}\in\mathbb{R}^m,\ \mathbf{f}:\mathbb{R}^{2m+1}\to\mathbb{R}^m,\end{aligned}\qquad (۱)$$

که در آن ماتریس ژاکوبین $\partial\mathbf{f}/\partial\dot{\mathbf{x}}$ تکین است. بسیاری از معادلات دیفرانسیلی- جبری که در مسائل عملی ظهور می‌یابند، دارای شکل ساده‌تری هستند که از جمله‌ی آن‌ها شکل هزنبرگ است اما پیش از معرفی شکل هزنبرگ لازم است خاصیت مهمی در معادلات دیفرانسیلی جبری به نام ایندکس[4] معرفی شود.

تعریف ۱ [۲]: معادله‌ی دیفرانسیلی- جبری غیر خطی (۱) دارای ایندکس ν است اگر ν کوچکترین تعداد مشتقات به صورت

$$\begin{aligned}&\mathbf{f}(t,\mathbf{x},\dot{\mathbf{x}})=\mathbf{0},\\&\frac{\mathrm{d}\mathbf{f}(t,\mathbf{x},\dot{\mathbf{x}},\ddot{\mathbf{x}})}{\mathrm{d}t}=\mathbf{0},\\&\vdots\\&\frac{\mathrm{d}^{\mu}\mathbf{f}(t,\mathbf{x},\dot{\mathbf{x}},\ldots,\mathbf{x}^{(\mu+1)})}{\mathrm{d}t^{\mu}}=\mathbf{0}\end{aligned}\qquad (۲)$$

باشد به طوری که (۲) امکان استخراج سیستم $\dot{x}(t)=\phi(x,t)$ را تنها با محاسبات جبری بدهد.

حال با در نظرگیری تعریف ایندکس، شکل کلی معادله‌ی دیفرانسیلی- جبری هزنبرگ با ایندکس ν قابل تعریف است [۲]:

$$\begin{aligned}&\mathbf{0}=\mathbf{f}_0(\mathbf{x}_1(t),t)\\&\dot{\mathbf{x}}_1(t)=\mathbf{f}_1(\mathbf{x}_1(t),\mathbf{x}_2(t),t)\\&\dot{\mathbf{x}}_2(t)=\mathbf{f}_2(\mathbf{x}_1(t),\mathbf{x}_2(t),\mathbf{x}_3(t),t)\\&\vdots\\&\dot{\mathbf{x}}_{\nu-1}(t)=\mathbf{f}_{\nu-1}(\mathbf{x}_1(t),\mathbf{x}_2(t),\ldots,\mathbf{x}_{\nu}(t),t)\end{aligned}\qquad (۳)$$

که در آن $\dfrac{\partial\mathbf{f}_0}{\partial\mathbf{x}_1}\dfrac{\partial\mathbf{f}_1}{\partial\mathbf{x}_2}\cdots\dfrac{\partial\mathbf{f}_{\nu-1}}{\partial\mathbf{x}_{\nu}}$ دارای رتبه‌ی کامل است. در (۳)، معادله‌ی نخست یعنی $\mathbf{f}_0(\mathbf{x}_1(t),t)=\mathbf{0}$ معادله‌ی جبری و متغیر $\mathbf{x}_{\nu}(t)$ متغیر جبری خوانده می‌شود. معادلات هزنبرگ با ایندکس بیشتر از یک نقشی محوری در تئوری معادلات دیفرانسیلی- جبری ایفا می‌کنند و معادلات دیفرانسیلی- جبری عمومی‌تر قابل بازنویسی به شکل هزنبرگ با همان ایندکس هستند [۱۱]. معادلات هزنبرگ به صورت خاص در سیستم‌های مکانیکی قابل توجه هستند. از جمله بسیاری از معادلات سیستم‌های دینامیکی مقید مثل سیستم‌های چند جسمی یا ربات‌های غیرهولونومیک به صورت هزنبرگ به دست می‌آیند [۱۲،۱۳].

[1] Spectral methods
[2] Legendre
[3] Chebychev
[4] Index

[Downloaded from joc.kntu.ac.ir on 2026-11-09]
[DOR: 20.1001.1.20088345.1391.6.3.4.0]

۳- حل عددی معادله‌ی هزنبرگ

معادله‌ی دیفرانسیلی- جبری شکل هزنبرگ را به صورت زیر بازنویسی می‌نماییم:

$$\begin{aligned}\mathbf{e}(t)&=\mathbf{f}_0(\mathbf{x}_1(t),t)\\ \dot{\mathbf{x}}_1(t)&=\mathbf{f}_1(\mathbf{x}_1(t),\mathbf{x}_2(t),t)\\ &\vdots\\ \dot{\mathbf{x}}_{\nu-1}(t)&=\mathbf{f}_{\nu-1}(\mathbf{x}_1(t),\mathbf{x}_2(t),\ldots,\mathbf{x}_\nu(t),t)\end{aligned} \qquad (۴)$$

در معادله‌ی (۴) اگر پس از چند گام تضمین کنیم $\mathbf{e}(t)\to 0$، آنگاه می‌توان نتیجه گرفت حل عددی به سمت خمینه‌ی قید همگرا می‌شود. برای تضمین $\mathbf{e}(t)\to 0$ روش‌های مختلفی وجود دارد. اولین روش حل معادله‌ی $\mathbf{e}(t)=0$ براساس متغیر جبری است. همان‌طور که می‌دانیم این کار برای معادلات دیفرانسیلی- جبری ایندکس یک انجام می‌شود و پاسخ مناسبی هم تولید می‌کند. اما برای ایندکس‌های بالاتر چون $\dfrac{\partial \mathbf{e}}{\partial \mathbf{x}_\nu}$ تکین است، امکان یافتن پاسخ وجود ندارد.

برای سیستم با ایندکس ν، با ν بار مشتق‌گیری، معادله‌ی دیفرانسیلی معمولی برای کل سیستم به دست می‌آید. یعنی داریم:

$$\det\left(\frac{\partial}{\partial \dot{\mathbf{x}}_\nu}\frac{\mathrm{d}^\mu \mathbf{g}}{\mathrm{d}t^\mu}\right)\neq 0$$

و لذا با تشکیل این معادله‌ی دیفرانسیلی معمولی، می‌توان با استفاده از روش‌های ضمنی، سیستم را حل کرد. این روش در واقع تشکیل و حل معادله‌ی زیر است:

$$\frac{\mathrm{d}^\mu \mathbf{e}}{\mathrm{d}t^\mu}=\mathbf{e}^{(\mu)}=\mathbf{0}. \qquad (۵)$$

مشکل روش، این است که با حل معادله‌ی (۵) برای متغیر جبری، دیگر دلیلی وجود ندارد که قید در پاسخ به دست آمده برقرار بماند و در واقع نمی‌توان تضمین نمود که $\mathbf{e}(t)$ نیز به صفر میل کند. دو اصلاح روی معادله‌ی (۵) به صورت زیر اعمال می‌شود:

۱) نیازی نیست مشتق‌گیری، ν بار انجام شود. در واقع همان‌طور که در مسأله‌ی ایندکس یک، بدون مشتق‌گیری (صفر بار مشتق‌گیری) با استفاده از روش ضمنی، معادله حل می‌شود، برای ایندکس ν نیز می‌توان $\nu-1$ بار مشتق گرفته و معادله را به صورت ضمنی حل نمود.

۲) برای آنکه تضمین نماییم قید در پاسخ برقرار می‌ماند، باید تضمین نماییم که $\mathbf{e}(t)$ و مشتقات آن، همگی به صفر میل می‌کنند.

در روش کنترل با استفاده از سطح لغزشی[1]، مسأله‌ای مشابه با آنچه هدف ماست، بررسی شده است. معادله‌ی سطح لغزش به گونه‌ای است که خطا و مشتقات آن همگی صفر می‌شوند. سپس از روی آن، ورودی کنترلی محاسبه می‌شود. برای توضیح بیشتر، معادله‌ی هزنبرگ ایندکس دو را در نظر می‌گیریم:

$$\begin{aligned}\mathbf{e}(t)&=\mathbf{f}_0(\mathbf{x}_1(t),t)\\ \dot{\mathbf{x}}_1(t)&=\mathbf{f}_1(\mathbf{x}_1(t),\mathbf{x}_2(t),t)\end{aligned} \qquad (۶)$$

با نگاه به (۶) دیده می‌شود که اگر در این معادله $\mathbf{x}_2(t)$ را به عنوان متغیر کنترلی ورودی در نظر بگیریم، آنگاه $\mathbf{e}(t)=\mathbf{f}_0(\mathbf{x}_1(t),t)$ را نیز می‌توان به عنوان خروجی در نظر گرفت و هدف تعیین $\mathbf{x}_2(t)$ به گونه‌ای است که $\mathbf{e}(t)$ به سمت صفر میل نماید. لذا از این دیدگاه معادله‌ی دیفرانسیلی- جبری ایندکس دو بسیار نزدیک به مسأله‌ی تنظیم در مهندسی کنترل است.

سطح لغزش، سطحی پایدار و معمولاً به صورت زیر در نظر گرفته می‌شود:

$$\begin{aligned}\mathbf{s}&\triangleq\left(\frac{\mathrm{d}}{\mathrm{d}t}+\lambda\right)^k\mathbf{e}\\ &=\left(\frac{\mathrm{d}^k}{\mathrm{d}t^k}+\alpha_{k-1}\frac{\mathrm{d}^k}{\mathrm{d}t^k}+\cdots+\alpha_1\frac{\mathrm{d}}{\mathrm{d}t}+\alpha_0\right)\mathbf{e}\\ &=\sum_{i=0}^{k}\alpha_i\mathbf{e}^{(i)}=\mathbf{0}.\end{aligned} \qquad (۷)$$

که در آن $\lambda>0$ و $\alpha_i=\binom{k}{i}\lambda^i$ است. البته در حالت کلی، ضرایب α_i می‌توانند از این قاعده پیروی نکنند بلکه تنها کافی است طوری انتخاب شوند که معادله‌ی سطح لغزش، پایدار باشد. این انتخاب پایداری را تضمین می‌کند. برای سیستم هزنبرگ با ایندکس ν، سطح لغزشی (۷) را برای $k=\nu-1$ در نظر می‌گیریم. از طرفی داریم:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{f}_0&=\frac{\partial \mathbf{f}_0}{\partial \mathbf{x}_1}\mathbf{f}_1+\frac{\partial \mathbf{f}_0}{\partial t}=\frac{\partial \mathbf{f}_0}{\partial \mathbf{x}_1}\mathbf{f}_1+\Delta_1\\ \frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathbf{f}_0&=\frac{\partial \mathbf{f}_0}{\partial \mathbf{x}_1}\left(\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_1}\mathbf{f}_1+\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_2}\mathbf{f}_2\right)\\ &\quad+\frac{d}{dt}\left(\frac{\partial \mathbf{f}_0}{\partial \mathbf{x}_1}\right)\mathbf{f}_1(t)+\frac{d}{dt}\frac{\partial \mathbf{f}_0}{\partial t}\\ &=\frac{\partial \mathbf{f}_0}{\partial \mathbf{x}_1}\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_2}\mathbf{f}_2+\Delta_2\\ &\vdots\\ \frac{\mathrm{d}^{\nu-1}}{\mathrm{d}t^{\nu-1}}\mathbf{f}_0&=\frac{\partial \mathbf{f}_0}{\partial \mathbf{x}_1}\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_2}\cdots\frac{\partial \mathbf{f}_{\nu-2}}{\partial \mathbf{x}_{\nu-1}}\mathbf{f}_{\nu-1}+\Delta_{\nu-1}\end{aligned}$$

که در آن Δ_i - ها به صورت زیر هستند:

$$\begin{aligned}\Delta_1&=\frac{\partial \mathbf{f}_0}{\partial t}\\ \Delta_{i+1}&=\frac{\mathrm{d}}{\mathrm{d}t}\Delta_i+\frac{d}{dt}\left(\prod_{j=1}^{i}\frac{\partial \mathbf{f}_{j-1}}{\partial \mathbf{x}_j}\right)\mathbf{f}_i\\ &\quad+\left(\prod_{j=1}^{i}\frac{\partial \mathbf{f}_{j-1}}{\partial \mathbf{x}_j}\right)\left(\prod_{j=1}^{i}\frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_j}\mathbf{f}_j+\frac{\partial \mathbf{f}_i}{\partial t}\right)\end{aligned}$$

[1] Sliding Mode Control

[Downloaded from joc.kntu.ac.ir on 2026-11-09]
[DOR: 20.1001.1.20088345.1391.6.3.4.0]

همان‌طور که دیده می‌شود، در Δ_{i-1}، متغیر $\mathbf{x}_i$ و مشتقات آن، وجود ندارد و لذا $\frac{\partial \Delta_{i-1}}{\partial \mathbf{x}_i}=\mathbf{0}$ است. همچنین برای $i=\nu$ داریم:

$$\begin{aligned}&\frac{\partial}{\partial \mathbf{x}_\nu}\left(\frac{d^{\nu-1}}{dt^{\nu-1}}\mathbf{f}_0\right)\\&=\frac{\partial}{\partial \mathbf{x}_\nu}\left(\frac{\partial \mathbf{f}_0}{\partial \mathbf{x}_1}\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_2}\cdots\frac{\partial \mathbf{f}_{\nu-2}}{\partial \mathbf{x}_{\nu-1}}\mathbf{f}_{\nu-1}+\Delta_{\nu-1}\right)\\&=\frac{\partial}{\partial \mathbf{x}_\nu}\left(\frac{\partial \mathbf{f}_0}{\partial \mathbf{x}_1}\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_2}\cdots\frac{\partial \mathbf{f}_{\nu-2}}{\partial \mathbf{x}_{\nu-1}}\mathbf{f}_{\nu-1}\right)+\underbrace{\frac{\partial \Delta_{\nu-1}}{\partial \mathbf{x}_\nu}}_{=0}\\&=\frac{\partial}{\partial \mathbf{x}_\nu}\left(\frac{\partial \mathbf{f}_0}{\partial \mathbf{x}_1}\frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_2}\cdots\frac{\partial \mathbf{f}_{\nu-2}}{\partial \mathbf{x}_{\nu-1}}\mathbf{f}_{\nu-1}\right)\neq 0\end{aligned}$$

لذا طبق قضیه‌ی تابع ضمنی، حل معادله‌ی ضمنی (۷) برای $\mathbf{x}_\nu$، دارای جواب است. البته استفاده از این معادله، تنها در صورتی که مشتق‌گیری تحلیلی امکان‌پذیر باشد، مناسب است. می‌توان با روش‌های عددی نیز، مشتقات را تقریب زد ولی به جای این کار، اگر از معادله‌ی گسسته‌سازی شده استفاده کنیم، راحت‌تر است. در ادامه روابط برای شکل گسسته‌سازی شده، آورده شده است.

به راحتی می‌توان نشان داد که معادله‌ی دیفرانسیلی- جبری هزنبرگ با ایندکس ν، پس از گسسته‌سازی به کمک یکی از روش‌های صریح، به صورت زیر قابل نمایش است:

$$\begin{aligned}\mathbf{0}&=\tilde{\mathbf{f}}_0(\mathbf{x}_1^n,t_n)\\\mathbf{x}_1^{n+1}&=\tilde{\mathbf{f}}_1(\mathbf{x}_1^n,\mathbf{x}_2^n,t_n)\\&\vdots\\\mathbf{x}_{\nu-1}^{n+1}&=\tilde{\mathbf{f}}_{\nu-1}(\mathbf{x}_1^n,\mathbf{x}_2^n,\ldots,\mathbf{x}_{\nu-1}^n,t_n)\end{aligned}\qquad(۸)$$

که در آن $\mathbf{x}_i^n=\mathbf{x}_i(t_n)$ است. معادله‌ی $\mathbf{0}=\tilde{\mathbf{f}}_0(\mathbf{x}_1^{n+\nu-1},t_{n+\nu-1})$ را می‌توان به صورت ترکیبی از توابع تنها بر اساس مقادیر x_i^n (مقادیر فعلی حالت) نوشت. در ادامه برای سادگی، معادلات را برای سیستم هزنبرگ به شکل (۸) با $\nu=3$ به دست می‌آوریم. ولی تعمیم آن به ایندکس‌های بالاتر، ساده است. داریم:

$$\begin{aligned}\mathbf{0}&=\tilde{\mathbf{f}}_0(\mathbf{x}_1^{n+\nu-1},t_{n+\nu-1})\\&=\tilde{\mathbf{f}}_0(\mathbf{x}_1^{n+2},t_{n+2})\\&=\tilde{\mathbf{f}}_0(\tilde{\mathbf{f}}_1(x_1^{n+1},x_2^{n+1},t_{n+1}),t_{n+2})\\&=\tilde{\mathbf{f}}_0(\tilde{\mathbf{f}}_1(\tilde{\mathbf{f}}_1(x_1^n,x_2^n,t_n),\\&\qquad\tilde{\mathbf{f}}_2(x_1^n,x_2^n,x_3^n,t_n),t_{n+1}),t_{n+2})\\&\triangleq\mathbf{e}_{n+2}\end{aligned}$$

در هر لحظه، مقادیر $\mathbf{x}_1^n$ و $\mathbf{x}_2^n$ در اختیار هستند و لذا در معادله‌ی به دست آمده، تنها $\mathbf{x}_\nu^n$ نامعلوم است. از طرفی معادله‌ی به دست آمده برای $\mathbf{x}_\nu^n$ دارای جواب است زیرا برای آن داریم:

$$\det\frac{\partial \mathbf{e}_{n+2}}{\partial \mathbf{x}_3^n}=\det\frac{\partial \tilde{\mathbf{f}}_0}{\partial \mathbf{x}_1^n}\frac{\partial \tilde{\mathbf{f}}_1}{\partial \mathbf{x}_2^n}\frac{\partial \tilde{\mathbf{f}}_2}{\partial \mathbf{x}_3^n}\neq 0\qquad(۹)$$

در انتها، معادله‌ی سطح لغزش را به صورت زیر تشکیل می‌دهیم:

$$\begin{aligned}\mathbf{s}&=(z-a)^{\nu-1}\mathbf{e}_n\\&=\mathbf{e}_{n+\nu-1}+\alpha_{\nu-2}\mathbf{e}_{n+\nu-2}+\cdots+\alpha_0\mathbf{e}_n\end{aligned}\qquad(۱۰)$$

که در آن داریم:

$$\alpha_i=\binom{\nu-1}{i}(-a)^i\qquad(۱۱)$$

همچنین z عملگری است که به صورت زیر تعریف می‌شود (تبدیل z):

$$z\mathbf{e}_n=\mathbf{e}_{n+1}$$

برای پایداری کافی است a عددی بین صفر و یک انتخاب گردد. لذا دستگاهی به شکل زیر به دست می‌آید:

$$\begin{aligned}&\tilde{\mathbf{f}}_0(\tilde{\mathbf{f}}_1(\tilde{\mathbf{f}}_1(\mathbf{x}_1^n,\mathbf{x}_2^n,t_n),\tilde{\mathbf{f}}_2(\mathbf{x}_1^n,\mathbf{x}_2^n,\mathbf{x}_3^n,t_n),t_{n+1}),t_{n+2})\\&+\alpha_1\tilde{\mathbf{f}}_0(\tilde{\mathbf{f}}_1(\mathbf{x}_1^n,\mathbf{x}_2^n),t_{n+1})+\alpha\tilde{\mathbf{f}}_0(\mathbf{x}_1^n,t_n)=\mathbf{0}\end{aligned}\qquad(۱۲)$$

که در آن تنها $\mathbf{x}_3^n$ مجهول است و طبق (۹)، معادله‌ی (۱۲) همواره برای $\mathbf{x}_3^n$ دارای پاسخ است. برای ایندکس‌های بالاتر نیز به راحتی می‌توان معادله‌ی مشابه را تشکیل داد.

لذا الگوریتم حل به این صورت خواهد بود:

۱- در لحظه‌ی صفر، مقادیر حالت معلوم هستند. با در نظر گرفتن $n=0$ معادله‌ی (۱۲) را تشکیل داده و مقدار $\mathbf{x}_\nu^0$ اصلاح می‌شود.

۲- از روی مقادیر به دست آمده در لحظه‌ی n، مقادیر حالت‌ها در $n+1$ را محاسبه می‌کنیم. توجه شود که تا قبل از همگرایی به خمینه‌ی قید، این مقادیر با مقادیر به دست آمده در تشکیل معادله‌ی (۱۲) در مرحله‌ی پیش، ممکن است متفاوت باشند و باید دوباره محاسبه شوند. اما پس از همگرایی به قید، نیاز به محاسبه‌ی مجدد نیست.

۳- با در نظر گرفتن $n+1$ به عنوان مقدار جدید برای n، محاسبه‌ی $\mathbf{x}_\nu^n$ از روی مقادیر جدید به دست آمده و معادله‌ی (۱۲) تکرار می‌شود.

۴- در صورتی که به پایان شبیه‌سازی رسیدیم، خارج شده و در غیر این صورت به مرحله‌ی ۲ بازگشت می‌کنیم.

۴- انتخاب ضرایب سطح لغزش

همان طور که گفته شد، برای همگرایی، در حالت پیوسته کافی است λ بزرگ‌تر از صفر و در حالت گسسته a بین صفر و یک انتخاب گردد. این دو عدد، سرعت همگرایی را تعیین می‌کنند. هرچه مقدار λ بزرگ‌تر انتخاب شود، سرعت همگرایی بیشتر می‌شود. از طرف دیگر مقادیر خیلی بزرگ، دستگاه را خیلی ناهموار خواهند کرد که مطلوب نیست. در یک روش گام متغیر، دستگاه نیاز به گام‌های خیلی کوچکی برای حل دارد و در روش‌های گام ثابت، موجب ناپایداری حل خواهد شد. در حالت کلی مقدار λ باید تابعی از خطای حل و سرعت

[Downloaded from joc.kntu.ac.ir on 2026-11-09]
[DOR: 20.1001.1.20088345.1391.6.3.4.0]

همگرایی مورد نیاز باشد. روند نزول خطای فاصله از منحنی قید، در حالت پیوسته به صورت زیر است:

$$\mathbf{e}(t) \approx \mathbf{e}_0 t^{\nu-1} \mathrm{e}^{\lambda t} \tag{۱۳}$$

در صورتی که بخواهیم پس از N گام، خطا به زیر مقدار مطلوب **tol** برسد، باید داشته باشیم:

$$\begin{aligned} &\mathbf{e}_0 (Nh)^{\nu-1} e^{-\lambda(Nh)} \le \mathbf{tol} \\ &\Rightarrow \lambda \ge \max_i \left\{ \frac{1}{Nh} \ln \frac{e_{i0}(Nh)^{\nu-1}}{tol_i} \right\} \end{aligned} \tag{۱۴}$$

البته معمولاً کاربران چنین نیازهایی ندارند و این اطلاعات را نیز وارد نمی‌کنند. برای به دست آوردن یک مقدار پیش‌فرض برای λ، می‌توان گفت همگرایی پس از ده تا صد گام، انتخاب مناسبی است. همچنین اگر زمان تقریبی نشست را $(4\lambda(\nu-1))^{-1}$ در نظر بگیریم بازه‌ی تقریبی زیر برای λ به دست می‌آید:

$$\frac{1}{400h(\nu-1)} < \lambda < \frac{1}{40h(\nu-1)} \tag{۱۵}$$

در حل‌های ارائه شده در بخش پنجم، مقدار $\lambda = \frac{1}{200h(\nu-1)}$ در نظر گرفته شده است. در روش‌های گام متغیر نیز باید از $h_{\max}$. در معادلات بالا استفاده شود. به همین ترتیب می‌توان a را در حالت گسسته به دست آورد. داریم:

$$e(n) \approx e_0 n^{\nu-1} a^{-n} \tag{۱۶}$$

و لذا با فرضیات قبلی نتیجه می‌شود:

$$\mathbf{e}_0 N^{\nu-1} a^N \le \mathbf{tol} \Rightarrow a \le \min_i \left\{ \sqrt[N]{\ln \frac{tol_i}{e_{0i} N^{\nu-1}}} \right\} \tag{۱۷}$$

۵- حل مسائل نمونه با استفاده از روش ارائه شده

نرم‌افزارهای مختلفی برای حل عددی معادلات دیفرانسیلی- جبری ارائه شده‌اند. در نرم‌افزار MATLAB با استفاده از تابع ode15s امکان حل معادلات دیفرانسیلی-جبری به شکل $\mathbf{M}(t,\mathbf{x})\dot{\mathbf{x}} = \mathbf{f}(t,\mathbf{x})$ امکان‌پذیر است. در صورتی که ماتریس $\mathbf{M}(t,\mathbf{x})$ تکین باشد، سیستم دیفرانسیلی- جبری خواهد بود. اما این تابع تنها قادر به حل مسائل ایندکس یک است و برای ایندکس‌های بالاتر قابل استفاده نیست [۱۴]. در نرم‌افزار Wolfram Methematica می‌توان از تابع NDSolve برای حل معادلات دیفرانسیلی- جبری کمک گرفت ولی برای معادلات با ایندکس بالاتر از یک تضمین همگرایی وجود ندارد [۱۵]. دیگر نرم‌افزار مطرح APMonitor است که توانایی حل معادلات دیفرانسیلی- جبری با هر ایندکسی را داراست. همچنین بسته‌های نرم‌افزاری مختلفی نیز در زمینه‌ی حل عددی معادلات دیفرانسیلی موجود هستند که کتابخانه‌هایی برای حل عددی معادلات دیفرانسیلی- جبری دارند. از آن جمله DASSAL/LAPACK ، RADAU و GAMD را می‌توان نام برد که همگی به زبان فورترن پیاده‌سازی شده‌اند. از بین آنها DASSAL/LAPACK مخصوص مخصوص حل معادلات دیفرانسیلی- جبری ایندکس یک است ولی دو مورد دیگر عمومی هستند.

مثال‌هایی که در این بخش ارائه شده‌اند همگی ایندکس بالاتر از یک دارند و لذا نمی‌توانیم از MATLAB برای مقایسه‌ی پاسخ‌ها استفاده کنیم. در مثال اول و دوم پاسخ دقیق سیستم در اختیار است و لذا برای آنها پاسخ روش عددی با پاسخ دقیق سیستم مقایسه شده است. در مثال دوم از نرم‌افزار APMonitor نیز برای مقایسه‌ی پاسخ استفاده شده است. برای مثال آخر نیز هیچ یک از نرم‌افزارها، پاسخ همگرا تولید نکردند ولی با استفاده از بسته‌های نرم‌افزاری پاسخ آنها محاسبه شد. میزان خطای دوری از قیدها نیز به عنوان یکی از معیار بررسی پاسخ ارائه شده است.

مثال ۱: سیستم خطی ایندکس-۲ زیر را در نظر بگیرید [16]:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} + \mathbf{g}(t) \\ \mathbf{0} = \mathbf{C}\mathbf{x} + r(t), \end{cases} \quad 0 \le t \le 10,$$

که در آن

$$\mathbf{A} = \begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ 1+2t \end{bmatrix}, \mathbf{C}^{\mathrm{T}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

$$\mathbf{g}(t) = \begin{bmatrix} -\sin(t) \\ 0 \end{bmatrix}, r(t) = -e^{-t} - \sin(t)$$

با شرایط اولیه‌ی $x_1(0) = 1$ و $x_2(0) = 0$. جواب دقیق این معادله به صورت زیر است:

$x_1(t) = e^{-t}, x_2(t) = \sin(t), z(t) = \dfrac{\cos(t)}{1+2t}$. در ادامه با استفاده از روش ارائه شده، این مسأله حل شده است. معادله‌ی سطح لغزش را به صورت زیر تعریف می‌کنیم:

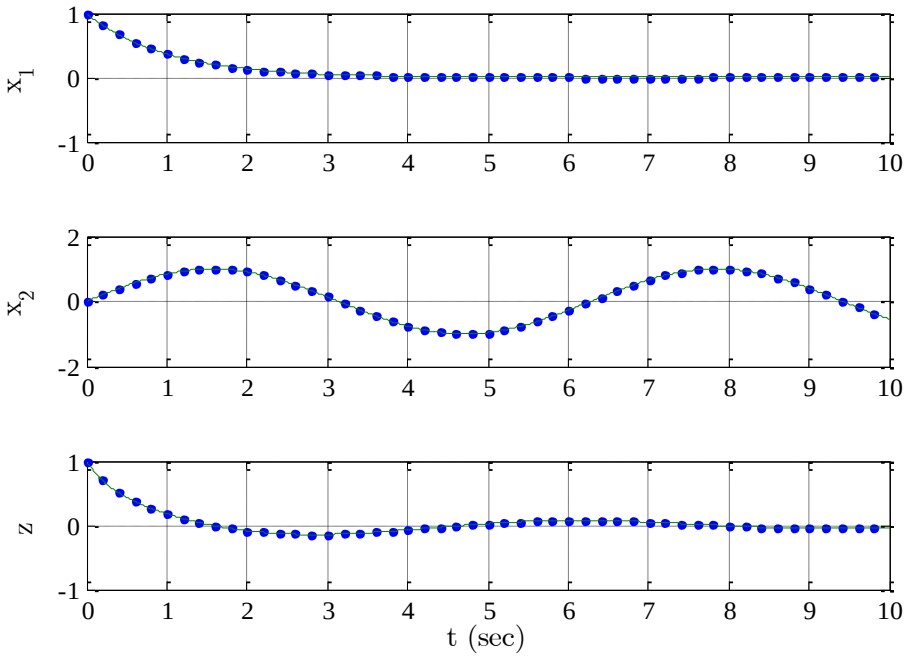
$$\begin{aligned} e = \mathbf{C}\mathbf{x} + r \Rightarrow s &= \left(\frac{\mathrm{d}}{\mathrm{d}t} + \lambda \right) e \\ &= \mathbf{C}\dot{\mathbf{x}} + \dot{r} + \lambda \mathbf{C}\mathbf{x} + \lambda r \\ &= \mathbf{C}\mathbf{A}\mathbf{x} + \mathbf{C}\mathbf{B}\mathbf{z} + \mathbf{C}\mathbf{g} + \dot{r} + \lambda \mathbf{C}\mathbf{x} + \lambda r \end{aligned}$$

و در نتیجه

$$\begin{aligned} z = &-(\mathbf{C}\mathbf{B})^{-1} \\ &(\mathbf{C}\mathbf{A}\mathbf{x} + \mathbf{C}\mathbf{g} + \dot{r} + \lambda \mathbf{C}\mathbf{x} + \lambda r) \end{aligned}$$

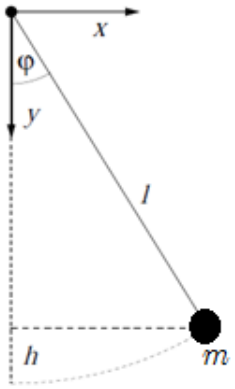
پاسخ این مسأله با استفاده از پیاده‌سازی روش ارائه شده در نرم‌افزار MATLAB به دست آورده شد. برای این مسأله طول گام حل را برابر $h = 10^{-2}$ و ضریب همگرایی $\lambda = (200h(\nu-1))^{-1} = 0.5$ در نظر گرفتیم. پاسخ دقیق و پاسخ عددی در شکل ۱ رسم شده است. برای نمایش بهتر، از هر ۲۰ نقطه تنها یک نقطه رسم شده است.

[Downloaded from joc.kntu.ac.ir on 2026-11-09]
[DOR: 20.1001.1.20088345.1391.6.3.4.0]



شکل ۱: پاسخ دقیق (منحنی پیوسته) و پاسخ عددی با استفاده از روش ارائه شده (نقطه‌چین)

مثال ۲: آونگ نشان داده شده در شکل ۲ را در نظر می‌گیریم [۱۷]. فرض شده است جرم m به یک میله به طول l متصل است.



شکل ۲- آونگ

برای به دست آوردن معادلات آونگ در فضای کارتزین، انرژی پتانسیل را برای آن می‌نویسیم:

$$U(x,y)=mgh=mgl-mgy \qquad (۱۸)$$

که در آن $\big(x(t),y(t)\big)$ مکان جرم m در لحظه‌ی t است. شتاب گرانش با g و ارتفاع آونگ با h نشان داده شده است. انرژی جنبشی نیز به صورت زیر به دست می‌آید:

$$T(\dot{x},\dot{y})=\frac{1}{2}m\left(\dot{x}^2+\dot{y}^2\right) \qquad (۱۹)$$

عبارت $\dot{x}^2+\dot{y}^2$ سرعت آونگ را مشخص می‌کند. همان‌طور که می‌دانیم جرم m آزادانه نمی‌تواند در هر نقطه‌ی دلخواه (x,y) باشد بلکه می‌تواند تنها روی دایره‌ای به شعاع l قرار گیرد. لذا طبق معادله‌ی دایره، قیدی به صورت زیر روی حالت‌های x و y خواهیم داشت:

$$0=g(x,y)=x^2+y^2-l^2 \qquad (۲۰)$$

با استفاده از (۱۸) تا (۲۰) تابع لاگرانژی را تشکیل می‌دهیم:

$$L(q,\dot{q})=T(\dot{x},\dot{y})-U(x,y)-\lambda g(x,y) \qquad (۲۱)$$

که در آن $\mathbf{q}=[x,y,\lambda]^T$ تعریف شده است. معادلات حرکت حال می‌توانند با استفاده از معادله‌ی اویلر به دست آیند:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_k}\right)-\frac{\partial L}{\partial q_k}=0,\quad k=1,2,3. \qquad (۲۲)$$

که به رابطه‌ی زیر می‌رسیم:

$$\begin{aligned} m\ddot{x}+2\lambda x&=0,\\ m\ddot{y}-mg+2\lambda y&=0,\\ g(x,y)&=0. \end{aligned} \qquad (۲۳)$$

با تعریف بردار $\mathbf{x}=[x,y,\dot{x},\dot{y},\lambda]^T$ معادلات را به صورت مرتبه‌ی اول می‌نویسیم:

[Downloaded from joc.kntu.ac.ir on 2026-11-09]
[DOR: 20.1001.1.20088345.1391.6.3.4.0]

$$\begin{aligned}\dot{x}_1 &= x_3\\ \dot{x}_2 &= x_4\\ \dot{x}_3 &= -\frac{2}{m}x_1x_5\\ \dot{x}_4 &= g-\frac{2}{m}x_2x_5\\ 0 &= x_1^2+x_2^2-l^2\end{aligned} \qquad (۲۴)$$

قابل ذکر است که سیستم (۲۴) به شکل هزنبرگ ایندکس سه است.

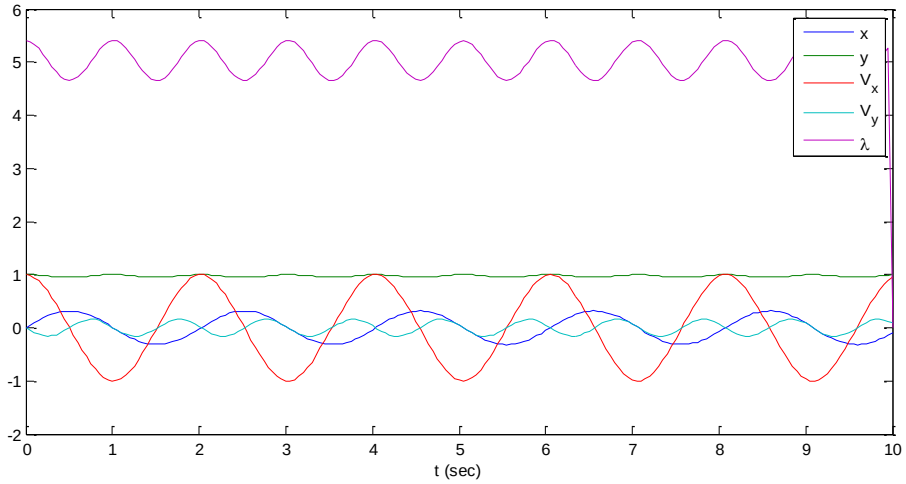
برای شبیه‌سازی و حل عددی، پارامترها به صورت زیر در نظر گرفته شده است:

$$m=1\text{kg},\quad l=1\text{m},\quad g=9.81\ \text{m/s}^2$$

همچنین فرض می‌کنیم آونگ از حالت عمودی، با سرعت اولیه یک متر بر ثانیه پرتاب شود. لذا شرایط اولیه به صورت زیر به دست می‌آید:

$$x_0=0,\quad y_0=l,\quad \dot{x}_0=1,\quad \dot{y}_0=0$$

در شکل زیر حل آن با استفاده از روش ردیاب خمینه‌ی قید آورده شده است:

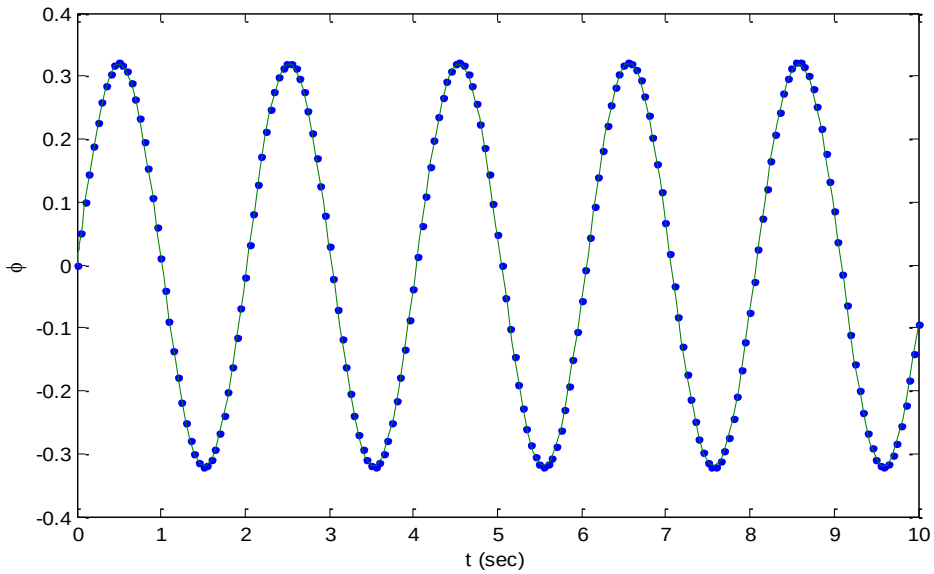


شکل ۳: حل عددی معادله‌ی آونگ با استفاده از روش ردیاب خمینه‌ی قید

برای بررسی درستی و دقت پاسخ، آن را با حل حاصل از مدل دیفرانسیلی معمولی آن مقایسه می‌کنیم. آونگ را می‌توان به صورت معادله‌ی دیفرانسیلی معمولی نیز مدل کرد. از روی پاسخ معادله‌ی دیفرانسیلی- جبری، می‌توان مقدار زاویه‌ی آونگ را محاسبه کرد:

$$\phi=\operatorname{atan2}(x,y)$$

در شکل ۴ حل معادله‌ی دیفرانسیلی- جبری با روش ردیاب خمینه‌ی قید و حل معادله‌ی دیفرانسیلی معمولی آونگ رسم شده‌اند:

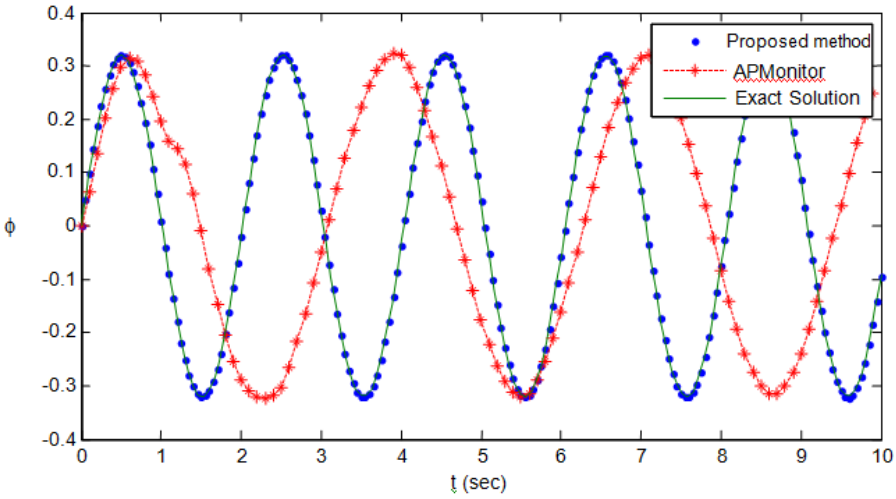


شکل ۴ : منحنی پیوسته از حل معادله‌ی دیفرانسیلی معمولی و نقاط از حل معادله‌ی دیفرانسیلی- جبری با استفاده از روش ارائه شده به دست آمده‌اند.

[Downloaded from joc.kntu.ac.ir on 2026-11-09]
[DOR: 20.1001.1.20088345.1391.6.3.4.0]

مقایسه با نرم‌افزارهای دیگر

در بین نرم‌افزارهای ارائه شده تنها APMonitor قادر به حل معادله‌ی دیفرانسیلی- جبری ایندکس سه است. در شکل ۵ نتیجه‌ی شبیه‌سازی آونگ با نرم‌افزار APMonitor آورده شده است:



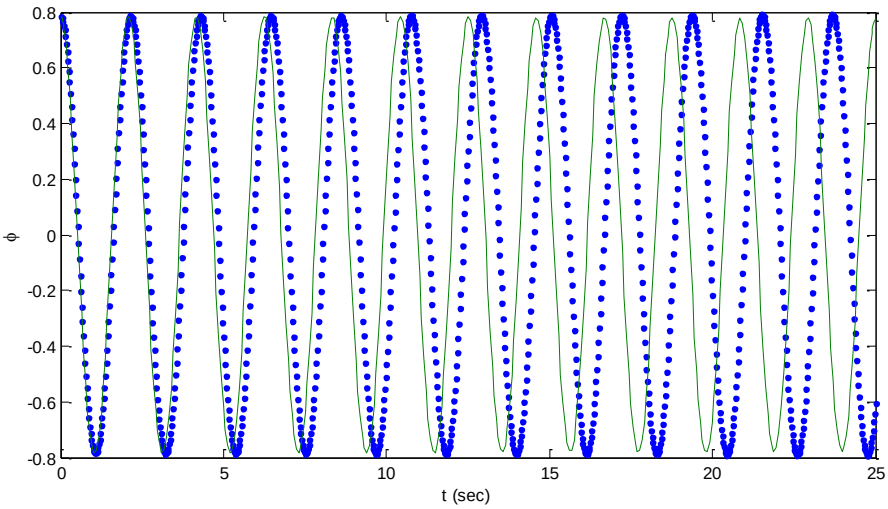
شکل ۵: حل دقیق معادله‌ی آونگ (نمودار پیوسته) در کنار حل عددی آن به وسیله‌ی روش ارائه شده در این مقاله (نقاط تو پر) و حل با استفاده از نرم‌افزار APMonitor (ستاره‌ها)

همان‌طور که در شکل ۵ دیده می‌شود، علاوه بر وجود اختلالاتی در زمان یک تا دو ثانیه، پاسخ عددی APMonitor به تدریج از پاسخ اصلی فاصله گرفته و نمی‌تواند قید را ارضاء نماید.

یکی از ویژگی‌های مهم روش ارائه شده، مقاوم بودن آن در مقابل شرایط اولیه‌ی ناسازگار است. برای بررسی این موضوع، در مثال قبل شرایط اولیه را در جهت عمود بر خمینه‌ی قید، دچار اختلال می‌کنیم. فرض می‌کنیم آونگ در زاویه‌ی ۴۵ درجه و بدون سرعت اولیه رها شود. شرایط اولیه‌ی ناسازگار را به صورت زیر در نظر می‌گیریم:

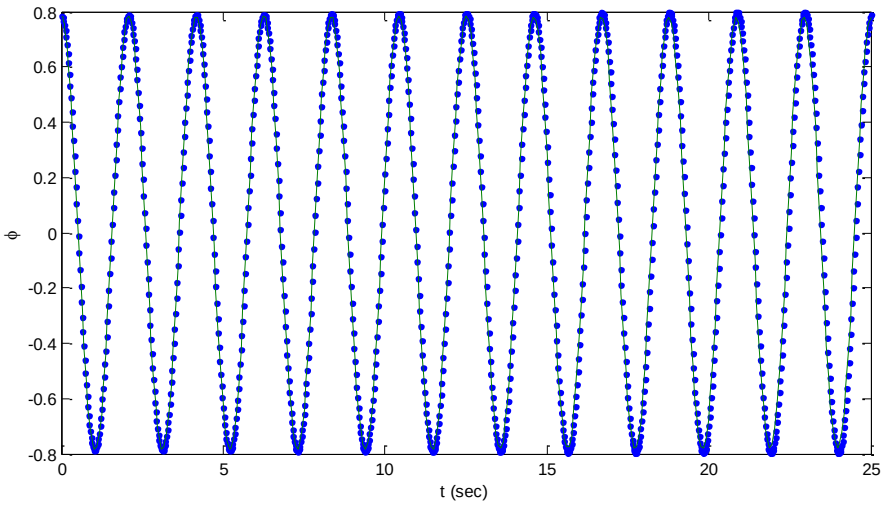
$$x_0 = 0.75,\ y_0 = 0.75,\ \dot{x}_0 = 0,\ \dot{y}_0 = 0$$

همچنین برای نمایش بهتر، مدت زمان شبیه‌سازی طولانی‌تر شده است. شکل ۶، پاسخ روش، بدون استفاده از کنترل خیمنه‌ی قید است که از حل معادله‌ی دیفرانسیلی که به کمک مشتق‌گیری از معادله‌ی دیفرانسیلی- جبری به دست آمده، رسم شده است.



شکل ۶: حل معادله‌ی دیفرانسیلی- جبری آونگ با شرایط اولیه‌ی ناسازگار بدون استفاده از روش ارائه شده

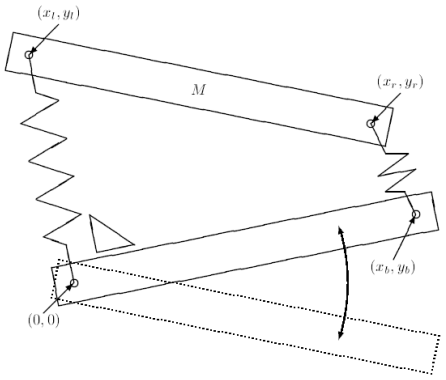
[Downloaded from joc.kntu.ac.ir on 2026-11-09]
[DOR: 20.1001.1.20088345.1391.6.3.4.0]



شکل ۷: حل معادله‌ی دیفرانسیلی- جبری آونگ با شرایط اولیه‌ی ناسازگار با استفاده از روش ارائه شده

همان‌طور که دیده می‌شود، در شبیه‌سازی‌های طولانی نیز، روش ارائه شده، به صورت مناسبی روی خیمنه‌ی قید باقی می‌ماند.

مثال ۳: مسأله‌ی محور ماشین [18]- در شکل ۸ مدل آن آورده شده است:



شکل ۸: مدل محور ماشین

شکل ۸ دو کارت را نشان می‌دهد که با دو فنر در انتها به یکدیگر متصل هستند. نقطه‌ای که با $(0,0)$ نشان داده شده است ثابت است و کارت پایینی می‌تواند حول آن بچرخد. نقطه‌ی (x_b, y_b) نیز به صورت کنترل‌شده حرکتی نوسانی دارد. وضعیت نقاط (x_r, y_r) و (x_l, y_l) خروجی مطلوب سیستم هستند. از آوردن نحوه‌ی محاسبه‌ی معادلات سیستم خودداری می‌شود و تنها معادله‌ی نهایی آورده شده است. مسأله به فرم زیر به دست می‌آید:

$$\begin{aligned} \dot{p} &= q, \\ K\dot{q} &= f(t,p,\lambda), \\ 0 &= \phi(t,p), \end{aligned} \qquad (۲۵)$$

که در آن $p,q \in \mathbb{R}^4$ ، $\lambda \in \mathbb{R}^2$ ، $0 \le t \le 3$ و تابع $f : \mathbb{R}^7 \to \mathbb{R}^4$ به صورت زیر است:

$$f(t,p,\lambda) = \begin{pmatrix} (L_0 - L_l)\dfrac{x_l}{L_l} + \lambda_1 x_b + 2\lambda_2 (x_l - x_r) \\ (L_0 - L_l)\dfrac{y_l}{L_l} + \lambda_1 y_b + 2\lambda_2 (y_l - y_r) - \varepsilon^2 \dfrac{M}{2} \\ (L_0 - L_r)\dfrac{x_r - x_b}{L_r} - 2\lambda_2 (x_l - x_r) \\ (L_0 - L_r)\dfrac{y_r - y_b}{L_r} - 2\lambda_2 (y_l - y_r) - \varepsilon^2 \dfrac{M}{2} \end{pmatrix} \qquad (۲۶)$$

همچنین داریم:

$$\begin{aligned} p &= [x_l, y_l, x_r, y_r], \\ L_l &= \sqrt{x_l^2 + y_l^2}, \\ L_r &= \sqrt{(x_r - x_b)^2 + (y_r - y_b)^2} \end{aligned}$$

تابع $\phi : \mathbb{R}^5 \to \mathbb{R}^2$ به صورت زیر است:

$$\phi(t,p) = \begin{pmatrix} x_l x_b + y_l y_b \\ (x_l - x_r)^2 + (y_l - y_r)^2 - L^2 \end{pmatrix}. \qquad (۲۷)$$

x_b و y_b نیز ورودی‌های تحریک سیستم هستند. این سیستم دارای ایندکس سه و به شکل هزنبرگ است. شکل معرفی شده‌ی (۲۵) یک شکل عمومی است که معمولاً در سیستم‌های چند جسمی به وجود می‌آید.

توابع تحریک سیستم را به صورت زیر در نظر می‌گیریم:

$$\begin{aligned} x_b(t) &= \sqrt{L^2 - y_b^2(t)}, \\ y_b(t) &= h\sin(\omega t). \end{aligned} \qquad (۲۸)$$

مقادیر ثابت نیز در زیر آورده شده‌اند:

$$\begin{aligned} &L = 1 && \varepsilon = 10^{-2} && h = 10^{-1} \\ &L_0 = 1/2 && M = 10 && \tau = \pi/5 \\ &\omega = 10 \end{aligned}$$

[Downloaded from joc.kntu.ac.ir on 2026-11-09]
[DOR: 20.1001.1.20088345.1391.6.3.4.0]

شرایط اولیه‌ی سازگار هم به صورت زیر داده شده است:

$$p_0=\begin{pmatrix}0\\1/2\\1\\1/2\end{pmatrix}, q_0=\begin{pmatrix}-1/2\\0\\-1/2\\0\end{pmatrix},$$

$$\dot{q}_0=\frac{2}{M\varepsilon^2}f(0,p_0,\lambda_0),$$

$$\dot{p}_0=q_0,\ \lambda_0=\dot{\lambda}_0=\begin{pmatrix}0\\0\end{pmatrix}.$$

قبل از شروع تابع f را به صورت زیر تعریف می‌کنیم:

$$\frac{1}{K}f(t,p,\lambda)=A(t,p)\lambda+B(t,p)$$

که در آن داریم:

$$A(t,p)=\frac{2}{M\varepsilon^2}\begin{bmatrix}x_b & 2(x_l-x_r)\\ y_b & 2(y_l-y_r)\\ 0 & -2(x_l-x_r)\\ 0 & -2(y_l-y_r)\end{bmatrix},$$

$$B(t,p)=\frac{2}{M\varepsilon^2}\begin{bmatrix}(L_0-L_l)\dfrac{x_l}{L_l}\\ (L_0-L_l)\dfrac{y_l}{L_l}-\varepsilon^2\dfrac{M}{2}\\ (L_0-L_r)\dfrac{x_r-x_b}{L_r}\\ (L_0-L_r)\dfrac{y_r-y_b}{L_r}-\varepsilon^2\dfrac{M}{2}\end{bmatrix}$$

همچنین در ادامه منظور از A_i یا B_i سطر iام این ماتریس‌ها هستند.

برای قید نخست (عمود بودن فنر سمت چپ) داریم:

$$\begin{aligned}e_1(t)&=x_lx_b+y_ly_b=p_1x_b+p_2y_b\\ \dot{e}_1(t)&=q_1x_b+p_1\dot{x}_b+q_2y_b+p_2\dot{y}_b\\ \ddot{e}_1(t)&=\dot{q}_1x_b+q_1\dot{x}_b+q_1\dot{x}_b+p_1\ddot{x}_b\\ &\quad+\dot{q}_2y_b+q_2\dot{y}_b+q_2\dot{y}_b+p_2\ddot{y}_b\\ &=(A_1x_b+A_2y_b)\lambda+B_1x_b+B_2y_b\\ &\quad+2q_1\dot{x}_b+2q_2\dot{y}_b+p_1\ddot{x}_b+p_2\ddot{y}_b\end{aligned}$$

برای قید دوم (ثابت بودن طول کارت بالایی) داریم:

$$\begin{aligned}e_2(t)&=(x_l-x_r)^2+(y_l-y_r)^2-L^2\\ &=(p_1-p_3)^2+(p_2-p_4)^2-L^2\\ \dot{e}_2(t)&=2(p_1-p_3)(q_1-q_3)\\ &\quad+2(p_2-p_4)(q_2-q_4)\\ \ddot{e}_2(t)&=2(q_1-q_3)^2+2(q_2-q_4)^2\\ &\quad(p_1-p_3)(A_1\lambda+B_1-A_3\lambda-B_3)+\\ &\quad(p_2-p_4)(A_2\lambda+B_2-A_4\lambda-B_4)\\ &=\begin{pmatrix}(p_1-p_3)(A_1-A_3)+\\(p_2-p_4)(A_2-A_4)\end{pmatrix}\lambda+\\ &\quad(p_1-p_3)(B_1-B_3)+\\ &\quad(p_2-p_4)(B_2-B_4)+\\ &\quad2(q_1-q_3)^2+2(q_2-q_4)^2\end{aligned}$$

لذا با تشکیل معادله‌ی سطح لغزش خواهیم داشت:

$$(\frac{d}{dt}+\lambda)^2=\ddot{e}(t)+\alpha_1\dot{e}(t)+\alpha_0e(t)=0.$$

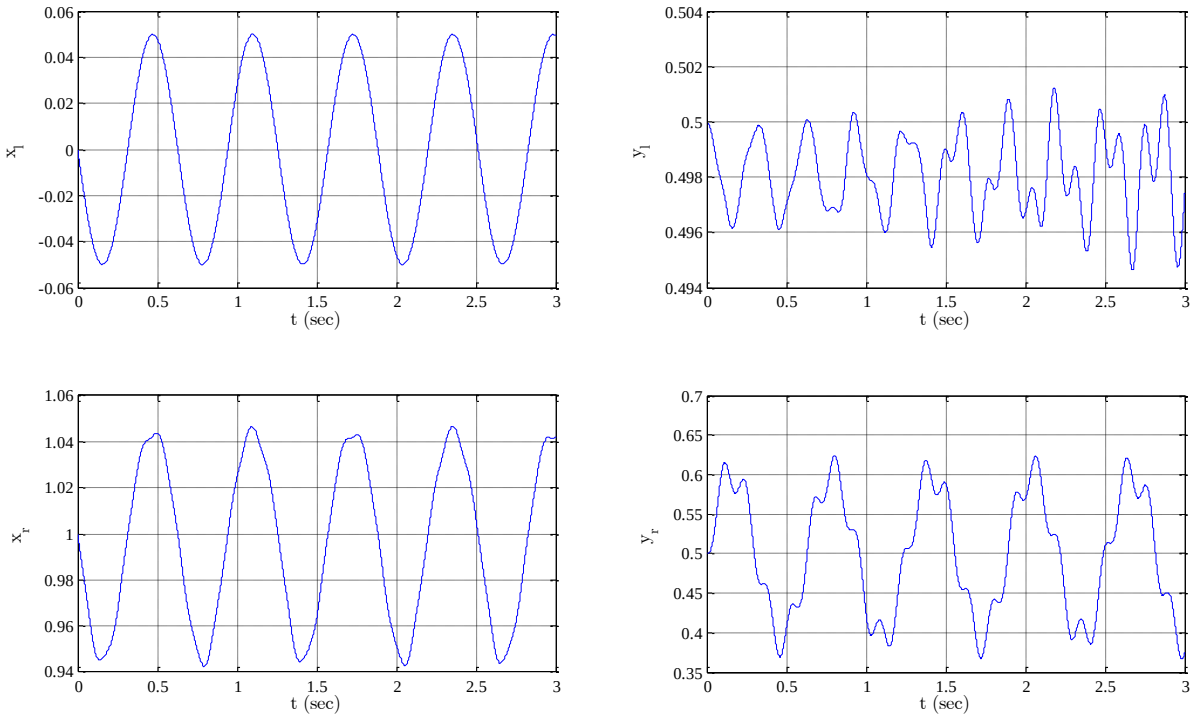
و در نتیجه:

$$\begin{aligned}0&=\begin{pmatrix}A_1x_b+A_2y_b\\(p_1-p_3)(A_1-A_3)\\+(p_2-p_4)(A_2-A_4)\end{pmatrix}\lambda\\ &+\begin{pmatrix}B_1x_b+B_2y_b+2q_1\dot{x}_b\\+2q_2\dot{y}_b+p_1\ddot{x}_b+p_2\ddot{y}_b\\(p_1-p_3)(B_1-B_3)+\\(p_2-p_4)(B_2-B_4)+\\2(q_1-q_3)^2+2(q_2-q_4)^2\end{pmatrix}\\ &+\alpha_1\begin{pmatrix}q_1x_b+p_1\dot{x}_b+q_2y_b+p_2\dot{y}_b\\2(p_1-p_3)(q_1-q_3)+\\2(p_2-p_4)(q_2-q_4)\end{pmatrix}\\ &+\alpha_0\begin{pmatrix}x_lx_b+y_ly_b\\(p_1-p_3)^2+(p_2-p_4)^2-L^2\end{pmatrix}\end{aligned}$$

که به راحتی در هر لحظه، برای λ قابل حل است.

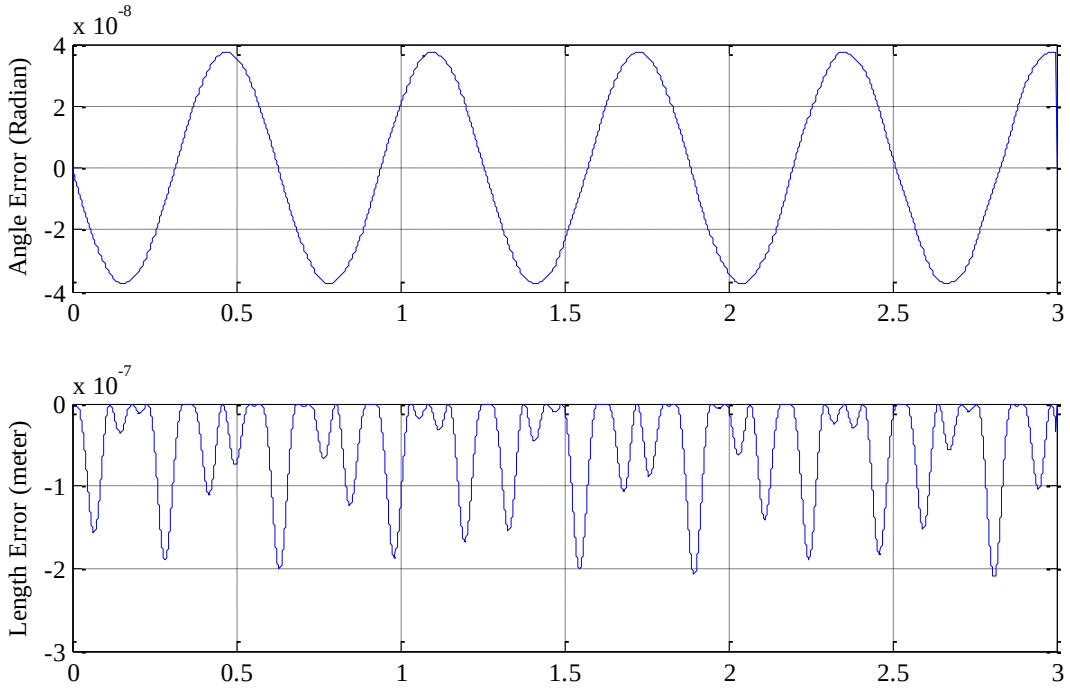
حل این سیستم بسیار دشوار است و اغلب نرم‌افزارهای موجود نتوانسته‌اند آن را حل کنند [۱۸]. اما این سیستم به کمک روش ارائه شده در این مقاله حل شده است. در شکل زیر نمودار (x_l,y_l) و (x_r,y_r) به عنوان خروجی‌های این سیستم رسم شده است:

[Downloaded from joc.kntu.ac.ir on 2026-11-09]
[DOR: 20.1001.1.20088345.1391.6.3.4.0]



شکل ۹- حل سیستم محور ماشین

برای بررسی درستی در شکل زیر وضعیت قیدهای سیستم نیز رسم شده است. سیستم دارای دو قید است. فنر سمت چپ عمود بر کارت پایینی نصب شده است لذا در حین شبیه‌سازی همواره زاویه‌ی بردار (x_l, y_l) با بردار (x_b, y_b) باید برابر ۹۰ درجه باشد. همچنین فاصله‌ی نقطه‌ی (x_l, y_l) از نقطه‌ی (x_r, y_r) باید برابر طول کارت بالایی یعنی یک متر باشد. در شکل ۱۰ خطای این قیدها در حین شبیه‌سازی رسم شده است.



شکل ۱۰- خطای فاصله از قیدها در شبیه‌سازی سیستم محور ماشین

[Downloaded from joc.kntu.ac.ir on 2026-11-09]
[DOR: 20.1001.1.20088345.1391.6.3.4.0]

همان‌طور که در شکل دیده می‌شود خطای زاویه از مرتبه‌ی 10^{-8} رادیان و خطای طول از مرتبه‌ی 10^{-7} متر است و نشان می‌دهد به شکل مناسبی قیود در حل عددی معادله‌ی دیفرانسیلی- جبری ارضاء شده‌اند.

مقادیر مرجع پاسخ در انتهای بازه‌ی انتگرال‌گیری $(t = 3\text{sec})$ به صورت زیر داده شده‌اند [۱۹]:

جدول ۱- پاسخ مرجع در پایان بازه‌ی انتگرال‌گیری

$x_l = 0.49345578427540280912 2\times10^{-1}$
$y_l = 0.496989460230171153861$
$x_r = 0.104174252488542151681\times10$
$y_r = 0.373911027265361256927$
$\dot{x}_l = -0.770583684040972357970\times10^{-1}$
$\dot{y}_l = 0.744686658723778553346 6\times10^{-2}$
$\dot{x}_r = 0.175568157537232222276\times10^{-1}$
$\dot{y}_r = 0.770341043779251976443$
$\lambda_1 = -0.473688659084893324729\times10^{-2}$
$\lambda_2 = -0.110468033125734368808\times10^{-2}$

در جدول ۲ وضعیت جواب حاصل از روش ارائه شده و نیز پاسخ بسته‌های RADAU و GAMD آورده شده است.

جدول ۲- مشخصات اجرا

روش حل	RADAU	GAMD	روش پیشنهادی
خطای نسبی	10^{-7}	10^{-7}	10^{-7}
خطای مطلق	10^{-7}	10^{-7}	10^{-7}
طول گام اولیه در نظر گرفته شده	10^{-7}	10^{-7}	10^{-7}
دفعات فراخوانی معادله حالت سیستم	2559	7167	1905
تعداد گام‌ها	289	98	335

از آن‌جا که پیاده‌سازی این دو نرم‌افزار به زبان فرترن بوده است به صورت کلی از پیاده‌سازی روش ارائه شده در این پروژه که به زبان MATLAB بوده است، سریع‌تر است. لذا برای مقایسه‌ی بهتر، تعداد دفعات فراخوانی تابع f (معادله‌ی حالت سیستم) و تعداد گام‌ها به عنوان معیارهای کارایی روش در نظر گرفته شده است. قابل ذکر است که در هر سه مورد، طول گام حل طوری تنظیم شده است که به دقت نسبی و مطلق 10^{-7} دست یابیم. همان‌طور که دیده می‌شود روش جدید به تعداد گام‌های بیشتری برای حل نیاز داشته است ولی تعداد دفعات فراخوانی معادله‌ی حالت سیستم (که همان تابع f است) کمتر است. در این پیاده‌سازی از رانگ-کوتای گام متغیر مراتب چهار و پنج برای انتگرال‌گیری ODE داخلی استفاده شده است. تعداد دفعات ایده‌آل فراخوانی تابع حالت در این روش، پنج برابر تعداد گام‌ها است که ممکن است در اثر ناموفق بودن طول گام انتخاب شده تکرار شود.

در حالت کلی توانایی در دستیابی به دقت مورد نظر کاربر، در کنار سرعت حل، می‌توانند معیارهای کارایی روش حل باشند. همان‌طور که پیش‌تر بیان شد بسیاری از روش‌ها قادر به حل معادلات ایندکس بالا نیستند ولی هر سه بسته‌ی نرم‌افزاری ارائه شده در جدول ۲ موفق به حل این مسأله شده‌اند. اما بررسی آن‌ها از نظر سرعت حل، تابعی از پارامترهای مختلفی است که به دست آوردن یک معیار سرراست برای مقایسه را دشوار می‌کند. حل معادله‌ی حالت، ریشه‌یابی تابع ضمنی یا محاسبه‌ی ماتریس ژاکوبین می‌توانند منابع اصلی هزینه‌ی محاسباتی و زمانی در مسائل مختلف باشند. لذا در نتیجه‌گیری عام‌تر، بسته به نوع مسأله روش‌های ارائه شده در جدول ۲ می‌توانند بر یکدیگر ارجحیت داشته باشند.

۸- نتیجه‌گیری

در این مقاله روشی برای حل معادلات دیفرانسیلی- جبری شکل هزنبرگ ارائه شد. ایده‌ی اصلی این روش براساس روش‌های حل معادلات دیفرانسیلی معمولی همراه با قید است. برای محاسبه و تضمین همگرایی پاسخ، یک سطح لغزشی پایدار تعریف شد و نشان داده شد که تشکیل سطح لغزشی متناسب با ایندکس سیستم، برای معادلاتی که به شکل هزنبرگ نوشته شده باشند، یک معادله‌ی پایدار برای محاسبه‌ی متغیر جبری در اختیار قرار می‌دهد. سپس برای مسائلی که امکان محاسبه‌ی مشتق تحلیلی امکان‌پذیر نیست، روش حل اصلاح شد و با گسسته‌سازی مستقیم معادله‌ی دیفرانسیلی، بدون نیاز به محاسبه‌ی مشتقات تحلیلی، مسأله حل شد. از این روش می‌توان برای حل معادلات هزنبرگ با ایندکس دلخواه استفاده نمود. در انتها حل عددی مثال‌های خطی و غیرخطی براساس این روش به دست آورده شد.

مراجع

[1] C. W. Gear and L. R. Petzold, "ODE methods for the solution of differential/algebraic systems," *SIAM Journal on Numerical Analysis*, vol. 21, no. 4, pp. 716–728, 1984.

[2] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical solution of initial-value problems in differential-algebraic equations*. Elsevier Science Publishing, 1989.

[3] U. M. Ascher and L. R. Petzold, *Computer methods for ordinary differential equations and differential-algebraic equations*. 1997.

[4] U. M. Ascher and L. R. Petzold, "Projected implicit Runge-Kutta methods for differential-

[Downloaded from joc.kntu.ac.ir on 2026-11-09]
[DOR: 20.1001.1.20088345.1391.6.3.4.0]

[11] R. Riaza, "Double SIB points in differential-algebraic systems," *IEEE Transaction on automatic control*, vol. 48, no. 9, pp. 1625–1629, 2003.

[12] P. Kunkel and V. Mehrmann, *Differential-algebraic equations analysis and numerical solution*. European Mathematical Society, 2006.

[13] P. Wijckmans, "Conditioning of differential algebraic equations and numerical solution pf multibody dynamics," 1996.

[14] "Solve stiff differential equations and DAEs; variable order method - MATLAB," 2012. [Online]. Available:

http://www.mathworks.com/help/matlab/ref/ode15s.html.

[15] "Numerical Solution of Differential-Algebraic Equations," 2012. [Online]. Available: http://reference.wolfram.com/mathematica/tutorial/NDSolveIntroductoryTutorialDAEs.html.

[16] F. Soltanian, S. M. Karbassi, and M. M. Hosseini, "Application of He's variational iteration method for solution of differential-algebraic equations," *Chaos, Solitons Fractals*, vol. 41, pp. 436–445, 2009.

[17] S. Schulz, "Four lectures on differential-algebraic equations," 2003.

[18] "Test set for IVP solvers." [Online]. Available: http://www.dm.uniba.it/~testset/problems/caraxis.php.

[19] F. Mazzia and C. Magherini, "Test set for initial value problem solvers, release 2.4," 2008.

algebraic equations," *SIAM Journal on Numerical Analysis*, vol. 28, pp. 1097–1120, 1991.

[5] E. Çelik and M. Bayram, "On the numerical solution of differential-algebraic equations by Pade series," *Applied Mathematics and Computation*, vol. 137, pp. 151–160, 2003.

[6] M. M. Hosseini, "Adomian decomposition method for solution of differential algebraic equations," *Journal of Computational and Applied Mathematics*, vol. 197, pp. 495–501, 2006.

[7] F. Soltanian, M. Dehghan, and S. M. Karbassi, "Solution of the differential algebraic equations via homotopy perturbation method and their engineering applications," *International Journal of Computer Mathematics*, vol. 87, no. 9, pp. 1950–1974, Jul. 2010.

[8] S. Karimi Vanani and A. Aminataei, "Numerical solution of differential algebraic equations using a multiquadric approximation scheme," *Mathematical and Computer Modelling*, vol. 53, pp. 659–666, Mar. 2011.

[9] H. Pasic, "An algorithm for a numerical solution of differential-algebraic equations," in *Proceedings of the 36th IEEE Conference on Decision & Control*, pp. 4912–4917.

[10] M. A. El-Khateb and H. S. Hussien, "An optimization method for solving some differential algebraic equations," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, pp. 1970–1977, May 2009.

[Downloaded from joc.kntu.ac.ir on 2026-11-09]
[DOR: 20.1001.1.20088345.1391.6.3.4.0]

[DOR: 20.1001.1.20088345.1391.6.3.4.0]
[Downloaded from joc.kntu.ac.ir on 2026-11-09]