مجله تحقیق در عملیات در کاربردهای آن، سال یازدهم، شماره سوم (پیاپی ۴۲)، پاییز ۹۳، ص ص ۱۳۷-۱۲۵
شاپا ۷۲۸۶-۲۲۵۱

حل دسته‌ای از مسایل کنترل بهینه با استفاده از الگوریتم ژنتیک ترکیبی

سعید نژادحسین*[۱]، عقیله حیدری[۲]

۱- دانشجوی دکترای، دانشگاه پیام نور، گروه ریاضی کاربردی، تهران، ایران

۲- دانشیار، دانشگاه پیام نور، گروه ریاضی کاربردی، مشهد، ایران

رسید مقاله: ۱ اسفند ۱۳۹۲

پذیرش مقاله: ۱ مرداد ۱۳۹۳

چکیده

در این مقاله با استفاده از پارامترسازی B -اسپلاین روشی جدید بر اساس الگوریتم ژنتیک ترکیبی اصلاح‌شده (MHGA) پیشنهادی، برای حل دسته‌ای از مسایل کنترل بهینه معرفی می‌کنیم. در هر گام از الگوریتم پیشنهادی، با افراز متساوی‌الفاصله‌ی بازه‌ی زمانی، MHGA با یک جمعیت اولیه‌ی کاملا تصادفی اجرا می‌شود، که هر فرد جمعیت یک ماتریس از مقادیر ورودی کنترل در گره‌های زمانی افراز جاری است. MHGA، الگوریتم GA را با SQP، به عنوان یک جستجوگر محلی ترکیب می‌کند. برای کاهش زمان محاسباتی، تعداد تکرارهای الگوریتم SQP، در گام‌های اولیه، کوچک انتخاب می‌شود و به تدریج، بعد از پیدا کردن ناحیه‌ی تقریبی جواب در فضای جستجو، در تکرارهای بعدی، آن را افزایش می‌دهیم. جواب بهینه‌ی حاصل از MHGA، به عنوان ضرایب B -اسپلاین در پارامترسازی کنترل استفاده می‌شود. برای تقریب بهتر منحنی کنترل در گام بعد، تعداد نقاط گره‌ای در افراز جاری را افزایش می‌دهیم و فرایند بالا را تکرار می‌کنیم. الگوریتم، زمانی متوقف می‌شود که نرم تفاضل دو منحنی کنترل پارامتری شده یا تفاضل مقادیر تابع هدف در دو گام متوالی از دقت‌های مطلوبی کم‌تر باشد. برای بررسی کارایی الگوریتم پیشنهادی، نتایج عددی روی بیشتر از ده مساله‌ی کنترل بهینه پیاده‌سازی شده است.

کلمات کلیدی: مساله‌ی کنترل بهینه، الگوریتم ژنتیک ترکیبی، پارامترسازی B -اسپلاین.

۱ مقدمه

مسایل کنترل بهینه یا بهینه‌سازی دینامیکی، کاربردهای زیادی در مسایل صنعتی نظیر مهندسی شیمی، رباتیک، هواپیمایی و غیره دارد [۱]. حل تحلیلی پیچیده‌ی این مسایل، به ارایه روش‌های عددی منجر می‌شود. دقت بالا و زمان محاسباتی کم، دو موضوع اصلی در این روش‌ها است. روش‌های حل عددی برای این مسایل در حالت کلی به دو دسته‌ی مستقیم و غیرمستقیم تقسیم می‌شود [۲].

* عهده‌دار مکاتبات

آدرس الکترونیکی: s_nejhadhosein@pnu.ac.ir

[Downloaded from jorlu.ir on 2026-11-29]

در روش‌های غیرمستقیم مساله‌ی کنترل بهینه با استفاده از حساب تغییرات به یک مساله‌ی مقدار مرزی چند نقطه‌ای تبدیل می‌شود که معمولا حل پیچیده‌ای دارد. بعلاوه این روش‌ها نیازبه اطلاعات گرادیان توابع و یک حدس اولیه‌ی خوب از جواب بهینه دارد؛ لذا ممکن است در بهینه‌ی محلی گرفتار شود. در روش‌های مستقیم مساله‌ی اصلی با گسسته‌سازی متغیرهای تصمیم، متغیرهای حالت یا/و کنترل، به یک مساله‌ی بهینه‌سازی متناهی‌البعد غیرخطی تبدیل می‌شود. دقت در این روش‌ها وابسته به نحوه‌ی گسسته‌سازی و پارامترسازی متغیرها است. معمولا گسسته‌سازی کم بازه‌ی زمانی به جواب‌های با دقت پایین در یک زمان محاسباتی کم و گسسته‌سازی زیاد منجر به دقت بالاتر ولی زمان محاسباتی بیشتر، به دلیل افزایش تعداد متغیرهای تصمیم منجر می‌شود. هرچند در برخی از مسایل، لزوما افزایش تعداد نقاط گره‌ای به بهبود کیفیت جواب منجر نمی‌شود [۳]. روش‌های تطبیقی نیز برای حل مستقیم مسایل به‌کار رفته است، که در این روش‌ها گسسته‌سازی بازه‌ی زمانی منطبق بر ساختار منحنی کنترل در هر گام تغییر می‌یابد. برخی از این روش‌ها در مقالات [۳،۴]، دیده می‌شود.

به جز روش‌های عددی کلاسیک بیان شده، الگوریتم‌های ابتکاری نیز در حل مسایل کنترل بهینه به‌کار رفته است. این دسته روش‌ها معمولا الگوریتم‌های بهینه‌سازی سراسری هستند که از عملگرهای تکاملی برای حل مساله استفاده می‌کنند. نمونه‌هایی از این روش‌های ابتکاری که در مقالات، برای حل مسایل کنترل بهینه به‌کار رفته، عبارت است از: الگوریتم ژنتیک (GA) [۵،۶،۷]، الگوریتم بهینه‌سازی تجمع ذرات (PSO) [۸،۹،۱۰]، بهینه‌سازی کلونی مورچگان (ACO) [۱۱]، الگوریتم تکامل دیفرانسیلی (DE) [۱۲،۱۳،۱۴] و الگوریتم برنامه‌ریزی ژنتیک (GP) [۱۵]. این روش‌ها به دلیل استفاده از ناحیه‌ی بزرگی از فضای جواب، عمدتا دارای زمان محاسباتی بالاتری نسبت به روش‌های عددی است.

در این مقاله با ترکیب یک روش عددی مستقیم و یک الگوریتم ابتکاری، یک الگوریتم جدید برای حل دسته‌ای از مسایل کنترل بهینه ارایه می‌دهیم. ایده‌ی روش بر اساس افزایش نقاط گسسته‌سازی، به صورت تدریجی، و کاهش زمان محاسباتی است. در واقع هدف رفع نواقص روش‌های ذکر شده از طریق ترکیب آن‌ها، به صورت یک الگوریتم تکراری، می‌باشد. روش مستقیم، از طریق پارامترسازی B –اسپلاین و الگوریتم ابتکاری از طریق یک الگوریتم ژنتیک ترکیبی اصلاح شده پیشنهادی، است. در هر گام، با افراز بازه‌ی زمانی به زیربازه‌های متساوی‌الفاصله، بر اساس تعداد نقاط پارامترسازی جاری، یک جمعیت اولیه‌ی تصادفی از مقادیر ورودی کنترل در این نقاط گره‌ای انتخاب می‌کنیم. برای کاهش زمان محاسباتی، تعداد نقاط در ابتدای الگوریتم کم انتخاب می‌شود. سپس جواب بهینه‌ی الگوریتم ژنتیک پیشنهادی را به عنوان ضرایب پارامترسازی B –اسپلاین برای منحنی‌های کنترل در نظر می‌گیریم. بدین ترتیب یک منحنی پارامتری از تابع کنترل، و در نتیجه منحنی حالت متناظر، به‌دست می‌آید. در ادامه برای بهبود کیفیت جواب به‌دست آمده، تعداد نقاط گره‌ای را برای گام بعدی افزایش می‌دهیم و فرایند بالا را، تا رسیدن به دقت مطلوب جواب، بر اساس خطاهای مرتکب شده در منحنی‌های پارامتری شده کنترل یا مقادیر تابع هدف متوالی، تکرار می‌کنیم. بدین ترتیب با افزایش تدریجی نقاط گره‌ای، دقت منحنی‌های جواب بهبود می‌یابد و زمان محاسباتی کم‌تر خواهد شد. ساختار مقاله به صورت زیر است: در بخش ۲، ابتدا فرمول‌بندی مسایل کنترل بهینه را بیان می‌کنیم. در بخش ۳، با معرفی الگوریتم ژنتیک و

[Downloaded from jorlu.ir on 2026-11-29]

عملگرهای آن، به معرفی الگوریتم ژنتیک ترکیبی اصلاح شده پیشنهادی، به عنوان یک زیر روال از الگوریتم اصلی می‌پردازیم. پارامترسازی B –اسپلاین را در بخش ۴ معرفی می‌کنیم. در بخش ۵، الگوریتم پیشنهادی را ارایه می‌دهیم. در انتها، در بخش ۶، به ارایه‌ی مثال‌های عددی برای بررسی کارایی الگوریتم پیشنهادی می‌پردازیم.

۲ مسایل کنترل بهینه

مسایل کنترل بهینه یا بهینه‌سازی دینامیکی (DO)، عبارت است از پیدا کردن یک ورودی کنترل برای سیستم، به‌طوری‌که ضمن برآوردن قیود دیفرانسیلی و فیزیکی، تابع هدف یا شاخص عملکرد سیستم را بهینه کند. انواع مختلف از این مسایل نظیر مساله‌ی حداقل زمان، مساله‌ی تعقیب، مساله‌ی ردیابی و غیره وجود دارد [۱]. در این مقاله ما یک مساله‌ی کنترلی زمان پیوسته‌ی غیرخطی را که در آن کنترل سیستم روی یک افق متناهی $[t_0,t_f]$ تعریف می‌شود، در نظر می‌گیریم. مساله عبارت است از پیدا کردن تابع برداری کنترل $u(0)\in\mathbb{R}^m$، که تابعک هزینه را با شرایط داده شده زیر مینیمم کند.

$$Min \quad J(u)=\phi(x(t_f),t_f)+\int_{t_0}^{t_f} g(x(t),u(t),t)dt \qquad (۱)$$

s.t.

$$\dot{x}=f(x(t),u(t),t), \qquad t_0\le t\le t_f \qquad (۲)$$

$$c_l(x(t),u(t),t)\le 0, \qquad l=1,2,...,n_c \qquad (۳)$$

$$d_k(x(t),u(t),t)\le 0, \qquad k=1,2,...,n_d \qquad (۴)$$

$$\Psi_h(x(t_f),t_f)=0, \qquad h=1,2,...,n_\Psi \qquad (۵)$$

$$x(t_0)=x_0 \qquad (۶)$$

که در آن $x(0)\in\mathbb{R}^n$ بردار حالت n–بعدی، t_0 و t_f به ترتیب زمان‌های اولیه و نهایی سیستم، g تابع اسکالر روی $\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}$، f,c_l,d_k توابع برداری روی $\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}$، Ψ_h–ها توابع برداری روی $\mathbb{R}^n\times\mathbb{R}$ و ϕ تابع اسکالر روی $\mathbb{R}^n\times\mathbb{R}$ می‌باشد. معادله‌ی (۱) شاخص عملکرد، (۲) معادلات حالت (۳)، معادلات و نامعادلات (۴) قیود فیزیکی، (۵) قیود حالت نهایی و معادله (۶) شرایط اولیه برای سیستم است.

۳ الگوریتم ژنتیک ترکیبی اصلاح شده

در این بخش یک الگوریتم ژنتیک ترکیبی اصلاح شده یا MHGA، را به عنوان یک زیر روال از الگوریتم اصلی معرفی می‌کنیم. در ابتدا الگوریتم ژنتیک و عملگرهای آن را تشریح می‌کنیم و سپس به معرفی نسخه‌ی ترکیبی اصلاح شده‌ی پیشنهادی آن می‌پردازیم.

[Downloaded from jorlu.ir on 2026-11-29]

۳-۱ الگوریتم GA

GA–را در سال ۱۹۷۵ هلند معرفی کرد، معروفترین الگوریتم‌های ابتکاری در حل مسایل بهینه‌سازی است. در این الگویتم‌ها با شروع از یک جمعیت اولیه از جواب‌های شدنی، در هر گام با استفاده از عملگرهای ژنتیکی، شامل تقاطع، جهش و ترکیب، جمعیت به‌روزرسانی می‌شود. در اینجا ما از GA با مولفه‌های زیر استفاده می‌کنیم.

جمعیت اولیه بازه‌ی زمانی $[t_0,t_f]$ را به N_t-1 زیر بازه‌ی متساوی‌الفاصله افراز می‌کنیم، که در آن N_t تعداد گره‌های زمانی است. مقادیر ورودی کنترل در هر گره زمانی به تصادف انتخاب می‌شود. این فرایند با مراحل زیر انجام می‌گردد.

۱- قرار دهید $t_i=t_0+ih,\ i=0,1,...,N_t-1$ که در آن $h=\frac{t_f-t_0}{N_t-1}$ طول گام است.

۲- مقدار ورودی کنترل در گره $(i+1)$-ام، t_i، برای $i=0,1,...,N_t-1$، عبارت است از یک بردار ستونی $m\times 1$، به صورت $u_i=\left[u_1^{(i)},u_2^{(i)},\dots,u_m^{(i)}\right]^T$، با مولفه‌های زیر

$$u_j^{(i)}=u_{left,j}+\left(u_{right,j}-u_{left,j}\right)r_{ij},\qquad j=1,2,...,m \tag{۷}$$

که در آن r_{ij} یک عدد تصادفی در بازه‌ی $[0,1]$ با یک توزیع یکنواخت است و $u_{left},u_{right}\in\mathbb{R}^m$ کران‌های بالا و پایین مقادیر ورودی کنترل است، که می‌تواند توسط کاربر، مساله (۱) در پیوست، یا توسط خود مدل، مساله (۴) در پیوست، تعریف شود. بنابراین هر فرد جمعیت یک ماتریس ورودی کنترل با بعد $m\times N_t$ است و عبارت است از $U=[u_k]_{k=0}^{N_t-1}$.

برازش برای هر ماتریس ورودی کنترل، U، متغیر حالت متناظر یک ماتریس $n\times N_t$ است، $X=[x_k]_{k=0}^{N_t-1}$، که در آن x_k، برای $k=0,1,...,N_t-1$، یک بردار ستونی $n\times 1$ است و به عنوان $(k+1)$-امین ستون ماتریس حالت در نظر گرفته می‌شود. مقادیر مولفه‌های ماتریس حالت را می‌توان با یک روش عددی حل معادلات دیفرانسیل روی سیستم حالت (۲)، با شرایط اولیه (۶)، به‌دست آورد. سپس تابع هزینه J، توسط یک روش انتگرال‌گیری عددی تقریب زده می‌شود، $\tilde{J}$. اگر DO شامل قیود تساوی، نامعادله یا شرط پایانی باشد، برای محاسبه‌ی برازش از روش جریمه استفاده می‌کنیم. نهایتا، ما مقدار $I(U)$، را به صورت زیر به عنوان برازش فرد جاری، تعریف می‌کنیم.

$$\begin{aligned}I(U)=\tilde{J}&+\sum_{k=1}^{n_d}\sum_{j=0}^{N_t-1}M_{1k}\max\left\{0,d_k(x_j,u_j,t_j)\right\}+\sum_{l=1}^{n_c}\sum_{j=0}^{N_t-1}M_{2l}c_l^2(x_j,u_j,t_j)\\&+\sum_{h=1}^{n_\Psi}M_{3h}\Psi_h^2(x_{N_t-1},t_{N_t-1})\end{aligned} \tag{۸}$$

که در آن M_{1k}، M_{2l} و M_{3h}–ها اعداد به اندازه کافی بزرگ است و $c_l(.,.,.),\ l=1,2,...,n_c$، $d_k(.,.,.),\ k=1,2,...,n_d$ و $\Psi_h(.,.),h=1,2,...,n_\Psi$ به ترتیب در (۳)، (۴) و (۵) تعریف شده است.

انتخاب با استفاده از عملگر مسابقه‌ی مرتبه‌ی ۸، دو والد را انتخاب می‌کنیم [۱۶].

تقاطع برای دو والد انتخابی، $U^{(1)}$ و $U^{(2)}$، گام‌های زیر را برای تولید فرزند جدید به‌کار می‌بریم.

[Downloaded from jorlu.ir on 2026-11-29]

۱- سه عدد تصادفی به صورت زیر انتخاب کنید.

$$\lambda_1 \in [0,1] \ , \ \lambda_2 \in [-\lambda_{\max} , 0] \ , \ \lambda_3 \in [1, 1+\lambda_{\max}] \qquad (۹)$$

که در آن $\lambda_{\max}$، یک عدد تصادفی در بازه‌ی $[0,1]$ است.

۲- قرار دهید

$$of^{\,k} = \lambda_k U^{(1)} + (1-\lambda_k) U^{(2)}, \qquad k = 1,2,3 \qquad (۱۰)$$

که در آن $\lambda_k , k = 1,2,3$، در (۹) تعریف شده است.

۳- قرار دهید $of = of^{\,*}$، که در آن $of^{\,*}$، بهترین $of^{\,k}$، برای $k = 1,2,3$، در (۱۰) می‌باشد.

جهش در اینجا جهش روی هر مولفه از فرزند طبق فرمول زیر به‌کار رفته است.

$$(of)_{ij} = (of)_{ij} + r_{ij}\,\beta, \qquad i = 1,2,...,m \ , \ j = 1,2,...,N_t \qquad (۱۱)$$

که در آن r_{ij} به صورت تصادفی در $\{-1, 1\}$ انتخاب می‌شود و β یک عدد تصادفی در بازه‌ی $[0,1]$ است. اگر $(of)_{ij} < u_{left,i}$ باشد آنگاه قرار می‌دهیم $(of)_{ij} = u_{left,i}$ و اگر $(of)_{ij} > u_{right,i}$ باشد آنگاه قرار می‌دهیم $(of)_{ij} = u_{right,i}$.

جایگزینی جایگزینی زمانی رخ می‌دهد که فرزند جدید دو خاصیت داشته باشد: اولا بهتر از بدترین فرد جمعیت باشد و ثانیا شبیه هیچ کدام از افراد نسل قبل نباشد.

شرط توقف الگوریتم زمانی متوقف می‌شود که حداقل یکی از شرایط زیر برقرار باشد: روی یک تعداد مشخصی از نسل‌ها، N_i، هیچ بهبودی رخ ندهد (بهترین فرد جمعیت تغییر نکند)، به حداکثر تعداد نسل برسیم، N_g، یا زمان اجرای الگوریتم، به عنوان یک پارامتر الگوریتم، $CPUTIME$، تمام شود.

۳-۲ الگوریتم MHGA

در این بخش MHGA، را به عنوان یک زیر روال الگوریتم اصلی تعریف می‌کنیم، که در آن از یک روش جستجوی محلی برای بهبود کیفیت جواب‌های GA استفاده شده است. در اینجا ما از روش SQP، به عنوان یک جستجوی محلی استفاده می‌کنیم. استفاده از SQP به عنوان یک جستجوگر محلی در روش‌های ابتکاری ترکیبی معمول است (برای مثال [۸] را ببینید). در شروع، ما از یک تعداد تکرار کم برای SQP استفاده می‌کنیم و در ادامه زمانی‌که یک ناحیه‌ی محتمل از فضای جستجو به‌دست آمد، ما تعداد تکرارها را به مرور افزایش می‌دهیم. این روش باعث کاهش زمان محاسباتی الگوریتم خواهد شد. فلسفه‌ی این روش در مرجع [۱۷] تشریح شده است. خلاصه‌ی الگوریتم MHGA در الگوریتم ۱ آورده شده است.

[Downloaded from jorlu.ir on 2026-11-29]

الگوریتم-۱ MHGA

مقداردهی اولیه مقادیر تعداد گره‌های زمانی، N_t، تعداد عناصر جمعیت، N_p، ماکزیمم تعداد نسل‌ها، N_g، ماکزیمم تعداد نسل‌ها بدون بهبود، N_i، حداکثر زمان اجرا، $CPUTIME$، احتمال جهش، P_m، کران‌های پایین و بالای مقادیر ورودی کنترل، u_{left} و u_{right}، مقدار اولیه برای حداکثر تکرار SQP، $sqp\max iter$ و یک جمعیت اولیه را وارد کنید.

ارزیابی مقدار برازش هر فرد را با استفاده از (۸) به‌دست آورید.

(حلقه‌ی تکرار) تا زمانی‌که شرط توقف بر قرار نشده است کارهای زیر را انجام دهید:

(انتخاب) دو والد $U^{(1)}$ و $U^{(2)}$ را بر اساس مسابقه‌ی ۸ نفره از بین عناصر جمعیت نسل جاری انتخاب کنید.

(تقاطع) با استفاده از والدین انتخابی گام قبل، یک فرزند جدید، با استفاده از (۹) و (۱۰)، ایجاد کنید.

(جهش) با احتمال P_m، و با استفاده از (۱۱) فرزند جدید را جهش دهید.

(جستجوی محلی) الگوریتم SQP، را با حداکثر تکرار $sqp\max iter$، روی فرزند جدید پیاده کنید.

(جایگزینی) اگر شرایط جایگزینی برقرار است، بدترین فرد جمعیت را با فرزند جاری جایگزین کنید.

قرار دهید $sqp\max iter = sqp\max iter + 1$.

خروجی بهترین فرد جمعیت نهایی را به عنوان جواب بهینه‌ی الگوریتم در نظر می‌گیریم.

٤ پارامترسازی B-اسپلاین

در تمام روش‌های مستقیم در حل مسایل DO، از یک پارامترسازی برای بردار کنترل استفاده می‌شود. در حالت کلی سه دیدگاه برای پارامترسازی در روش‌های مستقیم وجود دارد [۳]: روش پارامترسازی کنترل (CVP) یا دنباله‌ای، روش پرتابی چندگانه ، روش پارامترسازی کامل.

در اینجا ازروش CVP، بر اساس توابع B-اسپلاین استفاده شده است. این توابع دسته‌ای از اسپلاین‌ها [۱۸] است، که برای تقریب یا گسسته‌سازی توابع استفاده می‌شوند. B-اسپلاین‌ها روی یک مجموعه از نقاط گره‌ای تعریف می‌شود. فرض کنید $T = \{t_1, t_2, ..., t_n\}$، مجموعه‌ای از نقاط گره‌ای باشد، در این‌صورت، یک تابع B-اسپلاین مرتبه‌ی یک به صورت زیر تعریف می‌شود.

$$\varphi_j^{(1)}(t) = \begin{cases} 1 & , t_j \le t \le t_{j+1} \\ 0 & , o.w \end{cases} \quad , j = 1, 2, ..., n-1 \qquad (۱۲)$$

توابع B-اسپلاین مرتبه‌ی بالاتر $(1 < m = 2, 3, ..., n)$، به صورت بازگشتی زیر تعریف می‌شود.

$$\varphi_j^{(m)}(t) = \frac{t - t_j}{t_{j+m-1} - t_j} \varphi_j^{(m-1)}(t) + \frac{t_{j+m} - t}{t_{j+m} - t_j} \varphi_{j+1}^{(m-1)}(t) \quad , \quad j = 1, 2, ..., n-m \qquad (۱۳)$$

[Downloaded from jorlu.ir on 2026-11-29]

در حالت خاص، اگر $m = ۱$، یک پارامترسازی ثابت قطعه‌ای و اگر $m = ۲$ باشد، یک پارامترسازی خطی قطعه‌ای خواهیم داشت. هر بردار کنترل $u_i(t)$ به‌وسیله‌ی ترکیب خطی از این توابع، به صورت زیر تقریب زده می‌شود.

$$u_i(t) = \sum_{j=۱}^{n} \tilde{u}_{ij} \varphi_j^{(m)}(t), \quad i = ۱,۲,...,m \qquad (۱۴)$$

که در آن $\tilde{u}_{ij}$، برای $j = ۱,۲,...,n$، پارامترها یا ضرایب B-اسپلاین است. در حالت‌های خاص برای پارامترسازی قطعه‌ای ثابت یا خطی، این ضرایب برابر مقادیر ورودی کنترل در نقاط گره‌ای متناظر می‌باشد [۳].

۵ الگوریتم پیشنهادی

در این بخش با استفاده از الگوریتم MHGA و پارامترسازی B-اسپلاین یک الگوریتم جدید برای حل DO، ارایه می‌دهیم. در هر گام از الگوریتم پیشنهادی، تابع کنترل ورودی بر اساس پارامترسازی B-اسپلاین مرتبه‌ی یک یا دو، به‌روزرسانی می‌شود، که در آن پارامترهای B-اسپلاین، $\tilde{u}_{ij}$ در (۱۴)، از روی جواب بهینه‌ی MHGA، به‌دست می‌آید. مقدار تابع هدف بهینه در گام جاری را با $J^{*(k)}$ نمایش می‌دهیم. برای به‌دست آوردن جواب‌های با دقت بالاتر، در تکرار بعد، تعداد نقاط افراز بازه‌ی زمانی را دو برابر می‌کنیم. از آن‌جا که افزایش تعداد نقاط افراز، محاسبات بیشتری نیاز دارد، حداکثر زمان اجرای MHGA را بر اساس تعداد نقاط جدید افزایش می‌دهیم، یعنی داریم $CPUTIME = CPUTIME + N_t$.

الگوریتم زمانی متوقف می‌شود که نرم بینهایت اختلاف منحنی‌های کنترل پارامتری شده یا اختلاف مقادیر تابع هدف بهینه در دو گام متوالی از پارامترهای دقت، $\varepsilon_۱$ و $\varepsilon_۲$، کمتر باشد. خلاصه‌ی روش بیان شده، در الگوریتم ۲ آورده شده است.

الگوریتم-۲ الگوریتم پیشنهادی در حل DO
مقداردهی اولیه کران‌های u_{left} و u_{right}، پارامترهای دقت $\varepsilon_۱$، $\varepsilon_۲$ و مرتبه‌ی پارامترسازی، m، را وارد کنید. قرار دهید $k = ۱$. بردار ورودی اولیه، $u^{(۰)}$، را به دلخواه انتخاب کنید.
(حلقه‌ی تکرار) تا زمانی‌که $\left\|u^{(k)} - u^{(k-۱)}\right\|_\infty > \varepsilon_۱$ و $\left\|J^{*(k)} - J^{*(k-۱)}\right\| > \varepsilon_۲$ کارهای زیر را انجام دهید: ۱- MHGA را با یک جمعیت اولیه و پارامترهای N_t، N_p، N_g، N_i، $CPUTIME$، P_m و $sqp\max iter$ اجرا کنید. ۲- با استفاده از (۱۴)، پارامترسازی B-اسپلاین مرتبه‌ی m را با جواب بهینه گام قبل به‌دست آورید، $u^{(k)}$. ۳- تعداد نقاط افراز برای پارامترسازی منحنی کنترل را دو برابر کنید، $N_t = ۲N_t$. ۴- قرار دهید $CPUTIME = CPUTIME + N_t$.
خروجی جواب بهینه‌ی مساله‌ی DO، عبارت است از آخرین منحنی کنترل پارامتری شده، $u^*(t)$، منحنی حالت متناظر $x^*(t)$ و مقدار شاخص عملکرد بهینه‌ی متناظر، $J^* = J(u^*(t))$.

[Downloaded from jorlu.ir on 2026-11-29]

٦ نتایج عددی

در این بخش برای بررسی کارایی الگوریتم پیشنهادی، برخی مسایل عددی آورده شده است. نتایج با استفاده از نرم افزار Matlab 2011a، تحت سیستم عامل Windows 7 با CPU 2/53 MHz و 4 GB RAM به‌دست آمده است. هم‌چنین برای پیاده‌سازی SQP، در الگوریتم اصلی، از دستور fmincon، با تنظیم مناسب، استفاده کردیم. در اجرای الگوریتم پیشنهادی، ما روش انتگرال‌گیری عددی ذوزنقه‌ای مرکب را در تقریب انتگرال‌ها و روش رانگ-کوتا مرتبه ۴ را در حل عددی معادلات دیفرانسیل به‌کار بردیم. علاوه بر این ما دو نوع پارامترسازی قطعه‌ای ثابت و خطی، به ازای $m = ۱,۲$ در (۱۴)، را استفاده کردیم. در تمام مسایل از مقادیر پارامترهای ثابت $N_t = ۵$، $N_p = ۹$، $N_g = ۵۰$، $N_i = ۱۰$، $CPUTIME = ۵$، $p_m = ۰/۸$ و $sqp\max iter = ۴$ استفاده شده است.

ابتدا به تشریح دو مثال با جزئیات می‌پردازیم و سپس نتایج عددی مسایل آزمون را که در پیوست آمده، ارایه می‌دهیم.

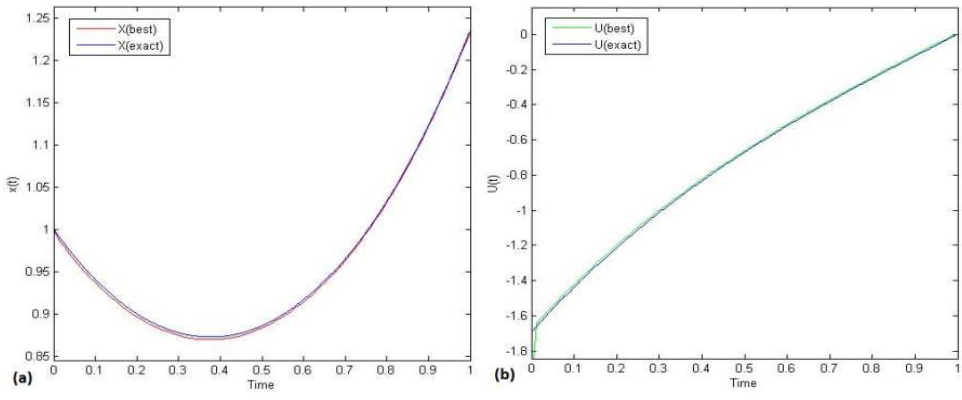
٦-١ مساله‌ی DO یک بعدی با جواب دقیق [۱۹] سیستم خطی اسکالر ساده‌ی زیر با یک متغیر کنترل و یک متغیر حالت، که دارای جواب دقیق است، مفروض است.

$$Min \quad J = \frac{۱}{۲}\int_{۰}^{۱}(x^{۲}+u^{۲})dt$$

s.t.

$$\dot{x} = x + u,$$
$$x(۰) = ۱.$$

شکل ۱، نمودارهای جواب بهینه‌ی تقریبی، با الگوریتم پیشنهادی، و جواب دقیق، شامل منحنی‌های کنترل و حالت را با هم مقایسه می‌کند. ملاحظه می‌شود این نمودارها تقریبا بر هم منطبق است.



شکل ۱. جواب‌های بهینه‌ی مساله ۶-۱: (a) منحنی‌های کنترل دقیق و تقریبی (b) منحنی‌های حالت دقیق و تقریبی

جدول ۱، نتایج عددی اجرای الگوریتم را در گام‌های متوالی برای پارامترسازی خطی قطعه‌ای، $m = ۲$، کران‌های $u_{left} = -۲$ و $u_{left} = ۰$ و دقت‌های $\varepsilon_۱ = ۰/۱$ و $\varepsilon_۲ = ۰/۰۱$ نشان می‌دهد. ستون‌ها در این جدول به

[Downloaded from jorlu.ir on 2026-11-29]

ترتیب گام، k، مقدار تابع هدف، $J^{(k)}$، اختلاف مقدار تابع هدف با جواب دقیق یعنی $J^{*}=۰/۸۴۴۷$، $E_{۱}^{(k)}$، نرم اختلاف منحنی‌های کنترل پارامتری شده در دو گام متوالی، $E_{۲}^{(k)}$، زمان محاسباتی گام جاری، $T^{(k)}$ و کل زمان محاسباتی $CT^{(k)}$ تا این مرحله را نشان می‌دهد. نتایج نشان می‌دهد مقادیر خطا در تکرارهای متوالی کاهشی است و این همگرایی روش را در این مثال تضمین می‌کند.

جدول ۱. نتایج عددی برای مساله‌ی ۶-۱

k	$J^{(k)}$	$E_1^{(k)}$	$E_2^{(k)}$	$T^{(k)}$	$CT^{(k)}$
۱	۰/۷۴۶۳	۰/۰۹۸۴	۲	۱/۲۰	۱/۲۰
۲	۰/۷۸۷۵	۰/۰۵۷۲	۰/۸۳۷۵	۳/۳۸	۴/۵۸
۳	۰/۸۱۵۵	۰/۰۲۹۲	۰/۷۵۰۸	۴/۹۴	۹/۵۲
۴	۰/۸۳۰۰	۰/۰۱۴۷	۰/۷۰۹۶	۴/۵۵	۱۴/۰۷
۵	۰/۸۳۷۴	۰/۰۰۷۳	۰/۶۹۰۶	۵/۹۲	۱۹/۹۹

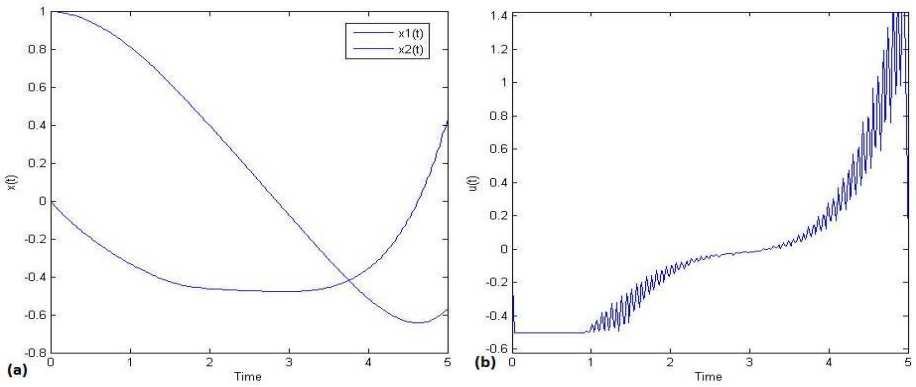
۶-۲ مساله‌ی واندرپل [۵] مساله واندرپل، شامل سیستم معادلات حالت با دو متغیر حالت، تابع هدف درجه دوم روی بازه‌ی زمانی [۰,۵] و یک شرط زمان پایانی است و به صورت زیر تعریف می‌شود.

$$Min \quad J=\frac{۱}{۲}\int_{۰}^{۵}(x_{۱}^{۲}+x_{۲}^{۲}+u^{۲})dt$$

s.t.

$$\dot{x}_{۱}=x_{۲},$$
$$\dot{x}_{۲}=-x_{۱}+(۱-x_{۱}^{۲})x_{۲}+u,$$
$$x_{۱}(۰)=۱\ ,\ x_{۲}(۰)=۰,$$
$$\psi_f=x_{۱}(۵)-x_{۲}(۵)+۱=۰.$$

الگوریتم پیشنهادی روی مساله پیاده‌سازی و نتایج در جدول ۲ خلاصه شده است. در شکل ۲، منحنی کنترل بهینه و منحنی‌های حالت‌های بهینه متناظر ترسیم شده است. مقادیر کران‌های پایین و بالای مقادیر ورودی کنترل به ترتیب $u_{left}=-۰/۵$ و $u_{right}=۲$ انتخاب شده است [۴]، علاوه بر این $\varepsilon_{۱}=۰/۱$ و $\varepsilon_{۲}=۰/۰۱$.



شکل ۲. نمودار جواب‌های بهینه‌ی مساله‌ی ۶-۲: (a) منحنی‌های حالت بهینه‌ی (b) منحنی کنترل بهینه

[Downloaded from jorlu.ir on 2026-11-29]

جدول۲ ، نتایج عددی پیاده‌سازی الگوریتم را به ازای تکرارهای متوالی نشان می‌دهد. ستون‌ها در این جدول به ترتیب گام، k، مقدار تابع هدف، $J^{(k)}$، مقدار شرط پایانی، $\psi_f^{(k)}$، زمان محاسباتی گام جاری، $T^{(k)}$، و کل زمان محاسباتی تا گام جاری، $CT^{(k)}$، را نشان می‌دهد. با توجه به نتایج عددی این جدول، مقادیر تابع هدف و شرایط پایانی در تکرارهای متوالی کاهشی است و این همگرایی روش در این مثال را تضمین می‌کند.

مساله‌ی واندرپل، در [۴]، با الگوریتم ژنتیک پیوسته (CGA)، حل شده است. نتایج با این روش عبارت است از $J^* = ۱/۷۴۰۴$ و $\psi_f = ۲/۶\times۱۰^{-۱۱}$. بعلاوه زمان محاسباتی در این روش ۵۰۱/۳۸ گزارش شده است. از مقایسه‌ی دو روش ملاحظه می‌کنیم روش پیشنهادی زمان محاسباتی را به صورت قابل توجه‌ای کاهش می‌دهد، هرچند مقدار شرط پایانی و تابع هدف در روش CGA، تا حدودی کم‌تر است.

جدول۲. نتایج عددی برای مساله‌ی ۶-۲

k	$J^{(k)}$	$\psi_f^{(k)}$	$T^{(k)}$	$CT^{(k)}$
۱	۴/۴۲۱۴	$۲/۰۱\times۱۰^{-۶}$	۲/۲۱	۲/۲۱
۲	۲/۵۷۵۴	$۲/۱۱\times۱۰^{-۱۰}$	۱/۸۸	۴/۰۹
۳	۱/۸۶۴۳	$۱/۹۱\times۱۰^{-۱۰}$	۱۵/۶۹	۱۹/۷۸
۴	۱/۷۵۸۲	$۱/۷۴\times۱۰^{-۹}$	۱۲/۷۶	۳۲/۵۴
۵	۱/۷۴۶۸	$۲/۷۷\times۱۰^{-۱۰}$	۶۴/۷۴	۹۷/۲۸
۶	۱/۷۴۲۱	$۱/۱۷\times۱۰^{-۱۰}$	۱۳۲/۸۵	۲۳۰/۱۳

۶-۳ مسایل آزمون

علاوه بر مساله‌های نمونه‌ی بالا، برای مقایسه‌ی انواع پارامترسازی قطعه‌ای خطی و ثابت، ۱۰ مساله آزمون دیگر در این بخش بررسی می‌شود. این مسایل به صورت خلاصه، به فرم (۱)-(۶) در بخش ۲، در پیوست آورده شده است. پارامترهای استفاده شده در هر مساله برای الگوریتم پیشنهادی و نتایج عددی مربوطه به تفکیک در جدول ۳ بیان شده است. این نتایج شامل مقدار بهینه‌ی تابع هدف، J^*، زمان محاسباتی مورد نیاز برحسب ثانیه، CT، و مقادیر حالت نهایی، $x(t_f)$، می‌باشد. در جدول ۳، نمادهای $P-C$ و $P-L$ برای دو نوع پارامترسازی، ثابت و خطی، استفاده شده است. نتایج نشان می‌دهد، با حفظ کیفیت جواب‌ها، زمان محاسباتی در پارامترسازی قطعه‌ای ثابت از خطی کم‌تر است.

جدول۳. نتایج عددی و پارامترهای الگوریتم پیشنهادی مربوط به مسائل آزمون ۱۰-۱ در پیوست

مساله	$(\varepsilon_1, \varepsilon_2, u_{left}, u_{right})$	پارامترسازی	J^*	CT	$x(t_f)$
۱	(۰/۱,۰/۰۱,−۱,۱)	$P-C$	۱/۰۴۸۲	۲۹/۵۴	$[۰/۱۰۰۲,-۰/۰۴۲۱]^T$
		$P-L$	۱/۰۸۳۳	۳۳۳/۷۷	$[۰/۰۴۹۹,-۰/۰۹۳۳]^T$
۲	(۰/۱,۰/۰۱,−۲۰,۲۰)	$P-C$	۰/۱۶۱۴	۵۷/۳۳	$[-۰/۲۱۸۰,۰/۰۱۱۴]^T$
		$P-L$	۰/۱۵۷۵	۵۸/۶۷	$[-۰/۲۱۰۹,۰/۰۲۵۶]^T$

[Downloaded from jorlu.ir on 2026-11-29]

مساله	$(\varepsilon_1, \varepsilon_2, u_{left}, u_{right})$	پارامترسازی	J^*	CT	$x(t_f)$
۳	(۰/۱,۰/۰۱,−۱.۵,۲)	$P-C$	۰/۰۱۶۰	۱۵/۳۱	$[-۱/۵۶\times۱۰^{-۹}, -۴/۱۵\times۱۰^{-۸}]^T$
		$P-L$	۰/۰۱۲۵	۱۵/۶۴	$[۷/۱۸\times۱۰^{-۱۲}, -۶/۵۹\times۱۰^{-۹}]^T$
۴	(۰/۱,۰/۰۱,−۱,۱)	$P-C$	$۹/۱۵\times۱۰^{-۴}$	۹/۷۶	$[۱/۹۱\times۱۰^{-۸}, -۳/۹۲\times۱۰^{-۹}]^T$
		$P-L$	$۹/۱۱\times۱۰^{-۴}$	۱۱/۲۰	$[۲/۵۸\times۱۰^{-۸}, ۱/۳۹\times۱۰^{-۸}]^T$
۵	(۰/۱,۰/۱,$-\pi,\pi$)	$P-C$	۳/۵۱۱۹	۱۱/۲۰	$[-۲/۹۶\times۱۰^{-۵}, ۴/۲۷\times۱۰^{-۶}]^T$
		$P-L$	۳/۴۹۱۱	۷۷/۳۶	$[۲/۵۴\times۱۰^{-۶}, -۳/۰\times۱۰^{-۷}]^T$
۶	(۰/۱,۰/۰۱,−۳,۳)	$P-C$	۲/۰۳۳۱	۲۷/۷۹	$[۴/۹۹\times۱۰^{-۱۱}, -۱]^T$
		$P-L$	۲/۰۷۰۹	۳۷/۲۹	$[-۲/۲۳\times۱۰^{-۹}, -۱]^T$
۷	(۰/۱,۰/۰۱,−۳۰,۳۰)	$P-C$	−۸/۸۶۹۲	۶/۰	$[-۱/۱۶\times۱۰^{-۹}, -۹/۸۵\times۱۰^{-۹}, ۸/۸۶۹۲]^T$
		$P-L$	−۸/۸۶۹۲	۶/۳۸	$[-۳.۷۹\times۱۰^{-۹}, -۷.۵۷\times۱۰^{-۹}, ۸.۸۶۹۲]^T$
۸	(۰/۱,۰/۰۱,−۱,۱)	$P-C$	۰/۰۳۵۹	۸۸/۰۶	$[۰/۹۶۵۲, ۰/۰۰۴۴, ۰/۰۹۴۱]^T$
		$P-L$	۰/۰۳۵۹	۱۲۲/۲۴	$[۰/۹۶۴۷, ۰/۰۰۴۴, ۰/۰۹۴۱]^T$
۹	(۰/۱,۰/۰۵,−۴,۴)	$P-C$	۳/۲۰۷۸	۲۹۶/۳۸	$[۴/۲۴\times۱۰^{-۸}, -۱/۶۸\times۱۰^{-۷}]^T$
		$P-L$	۳/۲۰۷۸	۳۳۴.۹۳	$[-۱/۳۵\times۱۰^{-۸}, -۸/۴۳\times۱۰^{-۹}]^T$
۱۰	(۰/۱,۰/۰۰۱,۰,۲)	$P-C$	$۴/۲۵\times۱۰^{-۱۴}$	۵.۷۳	۳/۹۷۴۳
		$P-L$	$۲/۱۲\times۱۰^{-۱۵}$	۶.۴۷	۳/۹۷۴۳

۷ نتیجه‌گیری

زمان محاسباتی و دقت عددی دو موضوع اساسی در حل مسایل DO، است. عمدتا روش‌های مستقیم، زمان محاسباتی کم و دقت پایین دارند و در مقابل روش‌های ابتکاری زمان محاسباتی بالاتر و دقت بهتری دارند. در این تحقیق با ترکیب روش‌های ابتکاری و مستقیم یک الگوریتم تکراری جدید برای حل ارایه شد، به طوری‌که در هر تکرار پارامترسازی متغیر کنترل بر اساس جواب بهینه‌ی یک الگوریتم ابتکاری پیشنهادی می‌باشد. نتایج عددی نشان می‌دهد روش ارایه شده زمان محاسباتی کمتری نسبت به الگوریتم ژنتیک، در مساله‌ی واندرپل، دارد و هم‌چنین در مقایسه‌ی دو پارامترسازی قطعه به قطعه ثابت و خطی، زمان محاسباتی پارامترسازی قطعه به قطعه ثابت کم‌تر است.

منابع

[۲] اسلامی راد، ا.، فراهی، م. ه.، (۱۳۹۲). تخمین پارامترهای مدل ریاضی HIV/AIDS با استفاده از مشاهده و حل مدل کنترلی به روش گسسته‌سازی. مجله تحقیق در عملیات درکاربردهای آن، ۱۰(۱)، ۸۷-۷۹.

[1] Kirk, D. E., (2004). Optimal Control Theory: An Introduction, Dover Publications.

[3] Schlegel, M., Stockmann, K., Binder, T., Marquardt, W., (2005). Dynamic optimization using adaptive control vector Parameterization. Computers Chemical Engineering, 29(8), 1731–1751.

[4] Assassa, F., Marquardt, W., (2014). Dynamic optimization using adaptive direct multipl shooting. Computers Chemical Engineering, 60(0), 242 – 259.

[Downloaded from jorlu.ir on 2026-11-29]

[5] Abo-Hammour, Z. S., Asasfeh, A. G., Al-Smadi, A. M., Alsmadi, O. M. K., (2011). A novel continuous genetic algorithm for the solution of optimal control problems. Optimal Control Applications and Methods, 32(4) 414-432.
[6] Sim, Y. C., Leng, S. B., Subramaniam, V., (2000). A combined genetic algorithms-shooting method approach to solving optimal control problems. International Journal of Systems Science, 31(1), 83-89.
[7] Shi, X. H., Wan, L. M., Lee, H. P., Yang, X. W., Wang, Y. C., Liang, Y. C., (2003). An improved genetic algorithm with variable population-size and a PSO-GA based hybrid evolutionary algorithm, in: Machine Learning and Cybernetics. International Conference on, Vol. 3, 1735-1740.
[8] Modares, H., Naghibi-Sistani, M. B., (2011). Solving nonlinear optimal control problems using a hybrid IPSO -SQP algorithm.Engineering Applications of Artificial Intelligence, 24(3), 476-484.
[9] Arumugam, M. S., Murthy, G. R., Loo, C. K., (2009). On the optimal control of the steel annealing processes as a two stage hybrid systems via PSO algorithms. International Journal Bio-Inspired Computing, 1(3), 198-209.
[10] Arumugam, M. S., Rao, M. V. C., (2008). On the improved performances of the particl swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems. Application Soft Computing 8(1), 324-336.
[11] van Ast, J. M., Babuˇska, R., De Schutter, B., (2009). Novel ant colony optimization approach to optimal control. International Journal of Intelligent Computing and Cybernetics, 2(3) 414-434.
[12] Cruz, I. L., Willigenburg, L. V., Straten, G. V., (2003). Efficient differential evolution algorithms for multimodal optimal control problems. Applied Soft Computing, 3(2) 97-122.
[13] Lee, M. H., Han, C., Chang, K. S., (1999). Dynamic optimization of a continuous polymer reactor using a modified differential evolution algorithm. Industrial and Engineering Chemistry Research, 38(12) 4825-4831.
[14] Wang, F. S., Chiou, J. P., (1997). Optimal control and optimal time location problems of differential-algebraic systems by differential evolution. Industrial and Engineering Chemistry Research, 36(12), 5348-5357.
[15] Kumar, A. V. A., Balasubramaniam, P., (2009). Optimal control for linear system using genetic programming.Optimal control applications and Methods, 30(1), 47-60.
[16] Engelbrecht, A., (2007). Computational Intelligence: An Introduction, Wiley.
[17] Babaie-Kafaki, S., Ghanbari, R., Mahdavi-Amiri, N., (2012). An efficient and practically robust hybrid metaheuristic algorithm for solving fuzzy bus terminal location problems. Asia-Pacific Journal of Operational Research, 29(2), 1-25.
[18] de Boor, C., (2001). A Practical Guide to Splines, no. v. 27 in Applied Mathematical Sciences.
[19] Chui, C. K., Chen, G., (1989). Linear Systems and Optimal Control . New York, NY, USA: Springer-Verlag New York, Inc.
[20] Fabien, B., C., (1998). Some tools for the direct solution of optimal control problem Advances Engineering Software, 29 , 45-61.

پیوست

مسایل نمونه آزمون طبق چارچوب مساله‌ی DO در (۶)-(۱)، بخش ۲، از مرجع [۲۰]، به صورت زیر است.

۱) $g=\frac{۱}{۲}(x_۱^۲+x_۲^۲+u^۲),\ t_۰=۰, t_f=۵, f=[x_۲,-x_۱+(۱-x_۱^۲)x_۲+u]^T,\ d=-(x_۲+۰/۲۵),$
$x_۰=[۱,۰]^T.$

۲) $g=x_۱^۲+x_۲^۲+۰/۰۰۵u^۲,\ t_۰=۰,\ t_f=۱, f=[x_۲,-x_۱+u]^T,\ d=-[-(۲۰+u)(۲۰-u),$
$-(۸(t-۰/۵)^۲-۰/۵-x_۲)]^T,\ x_۰=[۰,-۱]^T.$

۳) $g=\frac{۱}{۲}(x_۱^۲+x_۲^۲+۰/۱u^۲),\ t_۰=۰,\ t_f=۰/۷۸, f=[-۲(x_۱+۰/۲۵)+(x_۲+۰/۵)e^{\frac{۲۵x_۱}{x_۱+۲}}$
$-(x_۱+۰/۲۵)u, ۰/۵-x_۲-(x_۲+۰/۵)e^{\frac{۲۵x_۱}{x_۱+۲}}]^T, x_۰=[۰/۰۵,۰]^T, \psi=[x_۱,x_۲]^T.$

[Downloaded from jorlu.ir on 2026-11-29]

۴) $g=\frac{۱}{۲}(x_۱^۲+x_۲^۲), t_۰=۰, t_f=۰/۷۸, f=[-۲(x_۱+۰/۲۵)+(x_۲+۰/۵)e^{\frac{۲۵x_۱}{x_۱+۲}}-(x_۱+۰/۲۵)u,$

$۰/۵-x_۲-(x_۲+۰/۵)e^{\frac{۲۵x_۱}{x_۱+۲}}]^T, d=-(۱+u)(۱-u), x_۰=[۰/۰۵,۰]^T, \psi=[x_۱,x_۲]^T.$

۵) $g=\frac{۱}{۲}u^۲, t_۰=۰, t_f=۱۰, f=[\cos u-x_۲,\sin u]^T, d=-(\pi+u)(\pi-u), x_۰=[۳/۶۶,-۱/۸۶]^T,$

$\psi=[x_۱,x_۲]^T.$

۶) $\phi=x_۳, t_۰=۰, t_f=۱, f=[x_۲,u,\frac{۱}{۲}u^۲]^T, d=-(x_۱+۱/۹)(x_۱-۱/۹), x_۰=[۰,۰,۰]^T,$

$\psi=[x_۱,x_۲+۱]^T.$

۷) $\phi=-x_۳, t_۰=۰, t_f=۵, f=[x_۲,-۲+\frac{u}{x_۳},-۰/۰۱u], u, \frac{۱}{۲}u^۲]^T,$

$d=-(u+۳۰)(u-۳۰), x_۰=[۱۰,-۲,۱۰]^T, \psi=[x_۱,x_۲]^T.$

۸) $\phi=(x_۱-۱)^۲+x_۲^۲+x_۳^۲, g=\frac{۱}{۲}u^۲, t_۰=۰, t_f=۵, f=[x_۳\cos u, x_۳\sin u, \sin u]^T, x_۰=[۰,۰,۰]^T.$

۹) $g=\frac{۱}{۲}u^۲, t_۰=۰, t_f=۲, f=[x_۲,u]^T, x_۰=[۱,۱]^T, \psi=[x_۱,x_۲]^T.$

۱۰) $g=x^۲\cos^۲ u, t_۰=۰, t_f=\pi, f=\sin\frac{u}{۲}, x_۰=\frac{\pi}{۲}.$

[Downloaded from jorlu.ir on 2026-11-29]