



دانشکده علوم ریاضی  
گروه ریاضی کاربردی

پایان نامه کارشناسی ارشد

# یک کاربرد از شبکه‌های عصبی برای حل مسائل کنترل بهینه

رضوان کرمی

استاد راهنما

دکتر علیرضا ناظمی

شهریور ۱۳۹۴

## تقدیم به پدر و مادر عزیزم

آنان که همواره بر کوتاهی و درستی من، قلم عفو کشیده و گریانه از کنار غفلت هایم گذشته اند  
و در تمام عرصه های زندگی یار و یاور بی چشم داشت برای من بوده اند.

## تقدیم به برادرانم

که کنجینه های دلگرمی، مهربانی و محبت اند. کسانی که با فداکاری و گذشت مراد این امر خطیر  
همراهی و همیاری نمودند.

## سپاس‌گزاری...

سپاس‌گذاری را که به انسان توانایی و دانایی بخشد تا به بندگانش شفقت ورزد، مهربانی‌کننده و در حل مشکلات‌شان یاری‌شان نماید. از راحت‌خویش بگذرد و آسایش هم‌نوعان را مقدم دارد، با او معامله‌کننده و در این خلوص‌انبار نگیرد و خوش‌باشد که پروردگار سمیع است و بصیر...  
سپاس‌گذاری را که پدر و مادری فداکار نسیم ساخته تا در سایه درخت پر بار وجودشان بی‌سایم و از ریشه آن‌ها شاخ و برگ بگیرم و از سایه وجودشان در راه کسب علم و دانش تلاش‌نمایم. والدینی که بودنشان تاج افتخاری است بر سرم و نامشان دلیلی است بر بودنم. چرا که این دو وجود، پس از پروردگار مایه هستی ام بوده اند و راه رفتن را در این وادی پر از فراز و نشیب زندگی آموختند. آموزگاران‌ی که برایم زندگی، بودن و انسان‌بودن را معنا کردند...

سپاسگذار کسانی هستم که سر آغاز تولد من هستند، از یکی زاده می‌شوم و از دیگری جاودانه. استادی که سپیدی را بر تخته سیاه زندگیم نگاشت و مادری که تار موئی از او پای من سیاه‌ماند...

## تعمدنامه

اینجانب رضوان کرمی دانشجوی کارشناسی ارشد رشته ریاضی کاربردی دانشکده علوم ریاضی دانشگاه شاهرود، نویسنده پایان‌نامه با عنوان یک کاربرد از شبکه‌های عصبی برای حل مسائل کنترل بهینه، تحت راهنمایی دکتر علیرضا ناظمی متعهد می‌شوم:

- تحقیقات در این پایان‌نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش‌های دیگر پژوهش‌گران، به مرجع مورد استفاده استناد شده است.
- مطالب این پایان‌نامه، تاکنون توسط خود، یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ‌جا ارایه نشده است.
- حقوق معنوی این اثر، به دانشگاه شاهرود تعلق دارد، و مقالات مستخرج با نام “دانشگاه شاهرود” یا “Shahrood University” به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به‌دست آوردن نتایج اصلی پایان‌نامه تاثیرگذار بوده‌اند، در مقالات مستخرج از پایان‌نامه رعایت می‌گردد.
- در تمام مراحل انجام این پایان‌نامه، در مواردی که از موجود زنده (یا بافت‌های آنها) استفاده شده است، ضوابط و اصول اخلاقی رعایت شده است.
- در تمام مراحل انجام این پایان‌نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته (یا استفاده) شده است، اصل رازداری و اصول اخلاق انسانی رعایت شده است.

رضوان کرمی  
شهریور ۱۳۹۴

## مالکیت نتایج و حق نشر

- تمام حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه‌ای، نرم‌افزارها و تجهیزات ساخته شده) متعلق به دانشگاه شاهرود می‌باشد. این مطلب باید به نحو مقتضی، در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در این پایان‌نامه بدون ذکر منبع مجاز نمی‌باشد.

## چکیده

در این پایان نامه رده‌ای از مسائل کنترل بهینه غیرخطی با قیدهای مساوی و نامساوی را در نظر می‌گیریم و یک روش جدید بر اساس توانایی شبکه‌های عصبی مصنوعی در تقریب تابع ارائه می‌دهیم تا جواب مساله کنترل بهینه را بدست آوریم. برای انجام این کار سعی می‌کنیم شرایط هامیلتونین را بر پایه‌ی اصل مینیمم پونتریاگین (PMP) تقریب بزنیم، سپس بوسیله‌ی ساختن تابع خطا که شامل همه‌ی شرایط پونتریاگین است، یک مساله بهینه سازی نامقید را تعریف می‌کنیم. در تابع خطای پیشنهاد شده، از جواب‌های آزمایشی برای تابع مسیر، تابع کنترل و ضرایب لاگرانژ استفاده می‌کنیم. این جواب‌های آزمایشی بوسیله‌ی نرون‌ها ساخته می‌شوند. بنابراین، تابع خطا را که شامل وزن همه‌ی جواب‌های آزمایشی است را مینیمم می‌کنیم. با جایگذاری مقادیر بهینه وزن‌ها در جواب‌های آزمایشی، تابع بهینه مسیر، تابع بهینه کنترل و همچنین ضرایب لاگرانژ را بدست می‌آوریم.

**کلیدواژه:** کنترل بهینه، اصل مینیمم پونتریاگین، شبکه‌های عصبی، بهینه سازی.

# لیست مقالات مستخرج از پایان نامه

1. Karami. R & Nazemi. A, (2015), *Solving mixed optimal control problems using a feed forward neural network model*, 46<sup>th</sup> Annual Iranian Mathematics Conference, 616-619, Yazd.

۲. کرمی. ر، و ناظمی. ع، (۱۳۹۴)، ”حل مسائل کنترل بهینه با استفاده از یک مدل شبکه عصبی فیدفوروارد“، چهل و ششمین کنفرانس ریاضی ایران، ص ۶۳۴-۶۳۱، یزد.

# فهرست مطالب

خ	لیست تصاویر
۱	لیست جداول
۳	۱ مقدمات و مفاهیم اساسی
۳	۱.۱ مقدمه‌ای بر شبکه‌های عصبی
۳	۱.۱.۱ شبکه‌های عصبی بیولوژیکی
۴	۲.۱.۱ شبکه‌های عصبی مصنوعی
۹	۱.۲.۱.۱ شبکه‌های عصبی بازگشتی
۱۰	۳.۱.۱ شبکه‌های عصبی پیش‌خورد
۱۱	۴.۱.۱ تاریخچه تکامل شبکه عصبی
۱۳	۵.۱.۱ تاریخچه حل مسائل بهینه‌سازی با استفاده از شبکه‌های عصبی
۱۴	۶.۱.۱ شبکه عصبی (نرون مک کلاچ) برای تابع AND
۱۵	۷.۱.۱ شبکه عصبی (نرون مک کلاچ) برای تابع OR
۱۵	۸.۱.۱ شبکه عصبی (نرون مک کلاچ) برای تابع AND NOT
۱۶	۹.۱.۱ شبکه عصبی هب
۱۷	۱۰.۱.۱ شبکه عصبی آدالاین
۲۰	۱۱.۱.۱ شبکه عصبی مادالاین
۲۲	۱۲.۱.۱ شبکه‌های حافظه پیوندی
۲۲	۱۳.۱.۱ شبکه‌های عصبی هاپفیلد
۲۲	۱.۱۳.۱.۱ شبکه عصبی مصنوعی هاپفیلد
۲۴	۲.۱۳.۱.۱ نحوه عملکرد شبکه هاپفیلد پیوسته
۲۴	۳.۱۳.۱.۱ ساختار شبکه هاپفیلد پیوسته
۲۹	۲ مقدمه‌ای بر کنترل بهینه
۲۹	۱.۲ مقدمه
۲۹	۲.۲ تنظیم صورت مساله

۳۰	.....	مدل ریاضی	۱.۲.۲
۳۰	.....	محدودیت‌ها یا قيود فیزیکی	۲.۲.۲
۳۰	.....	تابعی معیار	۳.۲.۲
۳۱	.....	مساله‌ی کنترل بهینه	۳.۲
۳۲	.....	حساب تغییرات	۴.۲
۳۲	.....	تقریب توابع	۱.۴.۲
۳۳	.....	ساده ترین مساله تغییراتی	۲.۴.۲
۳۴	.....	شرایط لازم برای کنترل بهینه	۳.۴.۲
۳۵	.....	اصل حداقل یابی پونتریاگین	۵.۲
۳۷	.....	سیستم‌های تنظیم کننده‌ی خطی	۶.۲
۴۰	.....	مثال عددی	۷.۲
۴۳		شرایط بهینگی	۳
۴۳	.....	مقدمه	۱.۳
۴۳	.....	تعاریف اولیه	۲.۳
۴۵	.....	ساختار شبکه عصبی	۳.۳
۴۸	.....	مثال‌های عددی	۴.۳
۵۵		مسائل کنترل بهینه با قیدهای نامساوی	۴
۵۵	.....	مقدمه	۱.۴
۵۵	.....	بیان مساله	۲.۴
۵۷	.....	ساختار شبکه عصبی	۳.۴
۶۰	.....	مثال‌های عددی	۴.۴
۷۷		مراجع	
۸۳		واژه‌نامه فارسی به انگلیسی	
۸۵		واژه‌نامه انگلیسی به فارسی	



# لیست تصاویر

۴	.....	۱.۱ ساختار یک نرون.
۵	.....	۲.۱ مدل ریاضی یک نرون.
۶	.....	۳.۱ ساختار یک نرون با لایه ورودی، لایه پنهان و لایه خروجی.
۷	.....	۴.۱ شبکه عصبی مثال (۱.۱.۱)
۷	.....	۵.۱ تابع پله‌ای
۸	.....	۶.۱ تابع پله‌ای دو قطبی
۸	.....	۷.۱ تابع سیگموئید
۹	.....	۸.۱ تابع سیگموئید قطبی
۹	.....	۹.۱ شبکه‌های عصبی دو لایه بازگشتی.
۱۰	.....	۱۰.۱ ساختار شبکه عصبی مصنوعی (پرسپترون)
۱۱	.....	۱۱.۱ پرسپترون چندلایه
۱۵	.....	۱۲.۱ نرون مک کلاچ برای تابع AND
۱۵	.....	۱۳.۱ نرون مک کلاچ برای تابع OR
۱۵	.....	۱۴.۱ نرون مک کلاچ برای تابع AND NOT
۱۶	.....	۱۵.۱ شبکه عصبی هب
۱۸	.....	۱۶.۱ الگوریتم هب برای مثال (۲.۱.۱)
۱۹	.....	۱۷.۱ الگوریتم هب برای مثال (۲.۱.۱)
۲۰	.....	۱۸.۱ شبکه عصبی آدالین
۲۱	.....	۱۹.۱ شبکه عصبی مادالین
۲۲	.....	۲۰.۱ شبکه عصبی دیگر پیوندی
۲۲	.....	۲۱.۱ شبکه عصبی خود پیوندی
۲۳	.....	۲۲.۱ شبکه عصبی تشخیص الگو
۲۳	.....	۲۳.۱ ساختار شبکه هاپفیلد با سه نرون
۲۵	.....	۲۴.۱ مدل الکتریکی شبکه هاپفیلد
۲۵	.....	۲۵.۱ یک واحد از نرون شبکه هاپفیلد
۴۰	.....	۱.۲ دستگاه و کنترل کننده بازخورد بهینه برای مساله تنظیم کننده خطی

۴۴	.....	نرون‌های اولیه	۱.۳
۴۹	.....	تابع حالت برای مثال (۱.۴.۳)	۲.۳
۴۹	.....	تابع کنترل برای مثال (۱.۴.۳)	۳.۳
۵۰	.....	تابع حالت برای مثال (۲.۴.۳)	۴.۳
۵۱	.....	تابع کنترل برای مثال (۲.۴.۳)	۵.۳
۵۲	.....	تابع حالت برای مثال (۳.۴.۳)	۶.۳
۵۲	.....	تابع کنترل برای مثال (۳.۴.۳)	۷.۳
۵۴	.....	تابع حالت برای مثال (۴.۴.۳)	۸.۳
۵۴	.....	تابع کنترل برای مثال (۴.۴.۳)	۹.۳
۵۷	.....	پرسپترون دولایه با یک ورودی و یک خروجی	۱.۴
۶۲	.....	تابع حالت برای مثال (۱.۴.۴)	۲.۴
۶۲	.....	تابع کنترل برای مثال (۱.۴.۴)	۳.۴
۶۴	.....	تابع حالت برای مثال (۲.۴.۴)	۴.۴
۶۴	.....	تابع کنترل برای مثال (۲.۴.۴)	۵.۴
۶۶	.....	تابع حالت برای مثال (۳.۴.۴)	۶.۴
۶۶	.....	تابع حالت برای مثال (۳.۴.۴)	۷.۴
۶۷	.....	تابع کنترل برای مثال (۳.۴.۴)	۸.۴
۶۷	.....	مانع $\gamma$ و مرز آن که توسط دایره‌ها پوشیده شده است	۹.۴
۶۹	.....	مسیر بهینه تقریبی برای حالت اول برای مثال (۴.۴.۴)	۱۰.۴
۷۰	.....	مسیر بهینه تقریبی برای حالت دوم برای مثال (۴.۴.۴)	۱۱.۴
۷۰	.....	بدنه‌ی صلب ربات مسطح شناور آزاد	۱۲.۴
۷۲	.....	مسیر بهینه $x_1$ برای مثال (۵.۴.۴)	۱۳.۴
۷۲	.....	مسیر بهینه $x_2$ برای مثال (۵.۴.۴)	۱۴.۴
۷۳	.....	مسیر بهینه $x_3$ برای مثال (۵.۴.۴)	۱۵.۴
۷۳	.....	مسیر بهینه $x_4$ برای مثال (۵.۴.۴)	۱۶.۴
۷۴	.....	مسیر بهینه $x_5$ برای مثال (۵.۴.۴)	۱۷.۴
۷۴	.....	مسیر بهینه $x_6$ برای مثال (۵.۴.۴)	۱۸.۴

# لیست جداول

۱۰۱ توابع فعال سازی ..... ۱۲



# فصل ۱

## مقدمات و مفاهیم اساسی

### ۱.۱ مقدمه‌ای بر شبکه‌های عصبی

#### ۱.۱.۱ شبکه‌های عصبی بیولوژیکی

واحد اصلی دستگاه عصبی، سلولی خاص به نام نرون می‌باشد و بدون شک نحوه کار مغز و راز شعور آدمی در آن نهفته است. نرون‌ها با وجود تفاوت‌های زیاد از نظر اندازه و شکل ظاهری، مشخصه‌های مشترکی دارند. همان‌طور که در شکل (۱.۱) نشان داده شده است از تنه سلولی نرون تعدادی شاخک کوتاه خارج می‌شود که دندریت<sup>۱</sup> نام دارند. دندریت‌ها و تنه سلولی، سیگنال‌ها را از نرون‌های مجاور دریافت می‌کنند و از طریق یک لوله باریک به نام آکسون<sup>۲</sup> به نرون‌های دیگر منتقل می‌گردد. آکسون در انتهای خود به تعدادی رشته جانبی باریک تقسیم می‌گردد که پایانه‌های آکسونی نام دارد و با دندریت‌های سایر نرون‌ها مرتبط است. ارتباط بین آکسون یک نرون و دندریت نرون دیگر را سیناپس<sup>۳</sup> می‌نامند. یک سیناپس مرکب از پایانه قبل از سیناپسی<sup>۴</sup>، شکاف سیناپسی<sup>۵</sup> و پایانه بعد از سیناپسی<sup>۶</sup> می‌باشد. همه سیگنال‌های جمع‌بندی شده، در تنه نرون ترکیب می‌شوند و اگر وسعت سیگنال‌های ترکیب شده به آستانه نرون برسد، مرحله تحریک شدن<sup>۷</sup> فعال می‌شود و یک سیگنال خروجی تولید می‌شود. این سیگنال به صورت یک پالس منفرد یا بخشی از پالس‌ها در یک میزان خاص به موازات آکسون به پایانه‌های سیناپسی انتقال می‌یابد و موجب ترشح موادی به نام عصب-رسانه می‌شود. عصب-رسانه در داخل شکاف سیناپسی پخش می‌شود و نرون بعدی را تحریک می‌کند و به این ترتیب یک سیگنال از یک نرون به دیگری انتقال می‌یابد. تعداد بسیار زیادی آکسون از نرون‌های مختلف با دندریت‌های یک نرون به این صورت ارتباط برقرار می‌کنند. نرون‌ها با توجه به عملکردشان به سه دسته تقسیم می‌شوند:

(۱) نرون‌های حسی: این نرون‌ها پیام‌ها را از گیرنده‌ها به دستگاه عصبی مرکزی منتقل می‌کنند. گیرنده

<sup>۱</sup>Dendrite

<sup>۲</sup>Axon

<sup>۳</sup>Synapse

<sup>۴</sup>Presynaptic Terminal

<sup>۵</sup>Synaptic Cleft

<sup>۶</sup>Postsynaptic Terminal

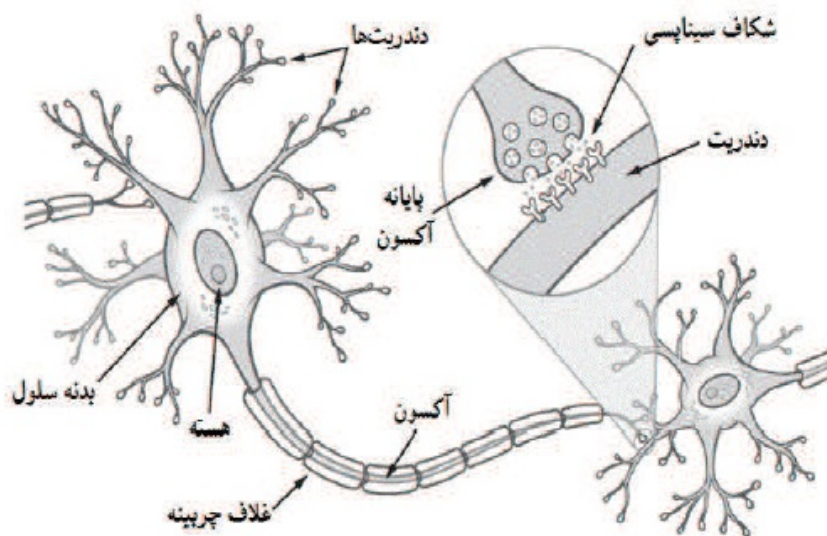
<sup>۷</sup>Firing

سلول خاصی است که در اندام‌های حسی، عضلات، پوست و ... قرار دارد و تغییرات فیزیکی یا شیمیایی را در می‌یابد، این تغییرات را به سیگنال‌های عصبی تبدیل می‌کند.

(۲) نرون‌های حرکتی: این نرون‌ها پیام‌ها را از مغز یا طناب نخاعی به اندام‌های عمل‌کننده می‌رسانند.

(۳) نرون‌های رابط: این نرون‌ها پیام‌ها را از نرون‌های حسی می‌گیرند و به یک نرون رابط دیگر انتقال می‌دهند و یا پیام‌ها را به یک نرون حرکتی می‌رسانند. نرون‌های رابط فقط در مغز، چشم و طناب نخاعی یافت می‌شوند.

حال منظور از یادگیری عبارت است از تغییری نسبتاً ساختاری در سیناپس (بعد از تحریک)، که این تغییر ساختاری باعث افزایش کارایی سیناپس می‌شود.

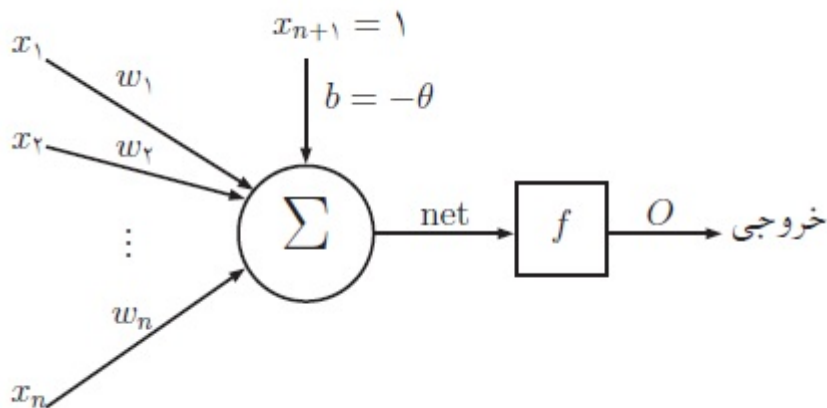


شکل ۱.۱: ساختار یک نرون.

## ۲.۱.۱ شبکه‌های عصبی مصنوعی

بعد از اشاره‌ای گذرا بر شبکه عصبی بیولوژیکی و نشان دادن گوشه‌ای بسیار کوچک از این جهان مرموز و فوق‌العاده پیچیده به موضوع اصلی یعنی شبکه‌های عصبی مصنوعی می‌پردازیم. شبکه‌های عصبی مصنوعی مدل‌های سخت افزاری یا نرم افزاری الهام گرفته از ساختار و رفتار نرون‌های بیولوژیکی و سیستم‌های عصبی انسان هستند که شامل عناصر پردازشی (به نام نرون‌ها) و ارتباطات میان آنها با ضرایب (وزن‌ها) اختصاص یافته به ارتباطات است، که ساختار شبکه عصبی را تشکیل می‌دهند. در این پایان‌نامه ما شبکه‌های عصبی مصنوعی را یک مدل ریاضی فرض خواهیم کرد و به جای استفاده از عبارت شبکه‌های عصبی مصنوعی به طور خلاصه عبارت شبکه‌های عصبی را به کار خواهیم برد.

مدل شبیه‌سازی شده یک نرون، در شکل (۲.۱) نمایش داده شده است. یک نرون در حالت کلی یک مجموعه از  $n + 1$  ورودی  $x_j$  ( $j = 1, \dots, n + 1$ ) دارد. این ورودی‌ها همان سیگنال‌های عصبی می‌باشند که از محیط یا از یک نرون دیگر فرستاده می‌شوند.  $w_j$  ( $j = 1, \dots, n + 1$ ) که روی آکسون‌ها



شکل ۲.۱: مدل ریاضی یک نرون.

واقع هستند نقش ساختار سیناپسی را ایفا می‌کنند. این‌ها هستند که باید تغییر کنند تا فرآیند یادگیری اتفاق افتد حال این‌که این تغییر بر چه قاعده‌ای استوار باشد بستگی به الگوریتم آموزش یا تعلیم دارد. وزن آخرین ورودی یعنی  $\theta$  را با  $b$  نشان می‌دهند و آن را بایاس<sup>۸</sup> می‌نامند. سیگنال‌های عصبی  $x_1$  و  $x_2$  و ... و  $x_{n+1}$  به ترتیب در  $w_1$  و  $w_2$  و ... و  $w_{n+1}$  ضرب شده و مجموع این حاصل ضرب‌ها که در شکل آن را با  $net$  نشان می‌دهیم، وارد نرون می‌شوند. در این مرحله نرون همانند یک تابع روی  $net$  عمل می‌کند و حاصل کار خود (برد تابع) را از طریق آکسون‌های طرف راست به نرون‌های دیگر می‌فرستد. این تابع را تابع فعال‌سازی<sup>۹</sup> می‌نامند.

البته خاطر نشان می‌کنیم که اگر نرون به‌عنوان تابع فرض شود این تابع از  $\mathbb{R}^n$  به  $\mathbb{R}$  تعریف می‌گردد. یعنی سیگنال خروجی از نرون در تمام آکسون‌های طرف راست یکسان می‌باشد که روی شکل با  $O$  نمایش داده‌ایم. همان‌طور که قبلاً اشاره شد نرون‌ها می‌توانند حسی، ارتباطی و یا حرکتی باشند. اطلاعات از طریق نرون‌های حسی دریافت می‌شود و از طریق نرون‌های ارتباطی انتقال می‌یابد و بالاخره نتیجه کار شبکه عصبی و یا واکنش شبکه عصبی به اطلاعات دریافت شده از طریق نرون‌های حرکتی به خارج منتقل می‌گردد (شکل (۳.۱)).

نرون‌های حسی را لایه ورودی، نرون‌های حرکتی را لایه خروجی و نرون‌های ارتباطی را لایه پنهان نیز می‌گویند. لایه مخفی می‌تواند از چندین لایه تشکیل گردد البته در شکل (۳.۱) لایه مخفی را تک لایه در نظر گرفته شده است.

نحوه‌ی اتصالات نرون‌ها می‌تواند بگونه‌ای باشد که شبکه تک لایه یا چندلایه بوجود آورد. شبکه‌های چند لایه از لایه ورودی که الگوهای ورودی به آن اعمال می‌شود و لایه خروجی که خروجی شبکه را تعیین می‌کند و لایه پنهان که که خود متشکل از چند لایه است و وظیفه‌ی ان ارتباط دادن لایه‌ی ورودی و لایه‌ی خروجی است، تشکیل می‌شود.

مشخصه‌های اصلی یک شبکه عصبی که در طراحی آن باید مدنظر قرار گیرند عبارت‌اند از :

<sup>۸</sup>Bias<sup>۹</sup>Activation Function

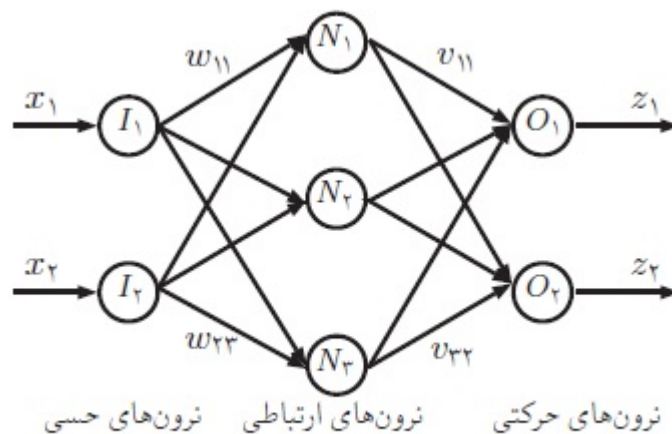
معماری شبکه عصبی: نحوه اتصالات بین نرون‌ها، تعداد آن‌ها و تعداد لایه‌های تشکیل دهنده بخش نرون‌های ارتباطی را معماری شبکه عصبی می‌گوییم.

تابع‌های فعال‌سازی: این‌که چه تابعی روی نرون قرار بگیرد تا بر ورودی‌های نرون اثر کرده و خروجی نرون را تولید کند تابع فعال‌سازی آن نرون گویند. در جدول (۱.۱) تعدادی از مهم‌ترین توابع فعال‌سازی نشان داده شده است.

الگوریتم آموزش: روشی که وزن‌های روی آکسون‌ها بر اساس آن تغییر می‌کنند ( $w_{ij}$  ها و  $v_{jk}$  ها در شکل (۳.۱)) تا خروجی شبکه عصبی به حالت مطلوب درآید را الگوریتم آموزش می‌گوییم.

با توجه به مطالب قبل می‌توان برای شبکه‌های عصبی مصنوعی خصوصیات عمومی زیر را نسبت داد:

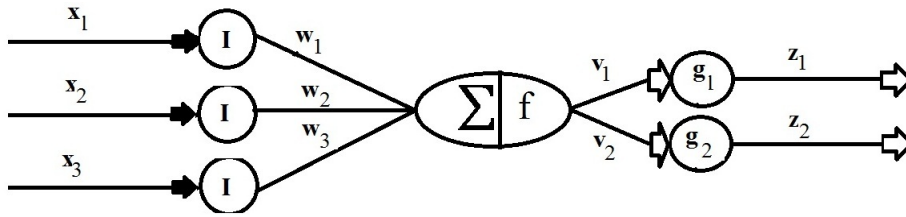
۱. پردازش اطلاعات در عناصر بسیار ساده به نام نرون (سلول عصبی) انجام می‌گیرد.
۲. سیگنال‌های عصبی بین سلول‌های عصبی از طریق اتصالات بین آن‌ها مبادله می‌شوند.
۳. به هر خط اتصال (آکسون) یک وزن نسبت داده می‌شود که در شبکه‌های عصبی معمولی این وزن‌ها در سیگنال‌های منتقل شده ضرب می‌گردند.
۴. هر نرون دارای یک تابع فعال‌سازی می‌باشد که این تابع در اکثر موارد غیرخطی است.



شکل ۳.۱: ساختار یک نرون با لایه ورودی، لایه پنهان و لایه خروجی.

مثال ۱.۱.۱.۱. می‌خواهیم خروجی شبکه زیر را تعیین کنیم: شبکه‌های عصبی مربوط به شکل را بصورت ریاضی بیان می‌کنیم. این شبکه در ۳ لایه ارائه شده است.





شکل ۴.۱: شبکه عصبی مثال (۱.۱.۱)

نورون‌های حسی سیگنال‌های دریافتی را عیناً انتقال می‌دهند این سیگنال‌ها در وزن‌های موجود در روی آکسون‌ها ضرب می‌شوند و در ورودی نورون ارتباطی با هم جمع می‌شوند.

$$w_1x_1 + w_2x_2 + w_3x_3,$$

اگر خروجی این لایه را با  $y$  نشان دهیم خواهیم داشت:

$$y = f(w_1x_1 + w_2x_2 + w_3x_3) = f\left(\sum_{i=1}^3 w_ix_i\right),$$

این خروجی هنگام عبور از آکسون‌های لایه بعدی باید در وزن‌های موجود در آن ضرب شود و بعد از گذر از لایه‌ی خارجی سیگنال‌های زیر به عنوان سیگنال حرکتی از شبکه عصبی خارج می‌گردند.

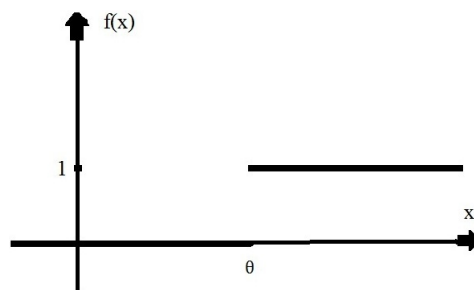
$$z_1 = g_1(v_1y), \quad z_2 = g_2(v_2y).$$

## توابع فعال سازی

چند نمونه از این توابع را در ذیل بیان می‌کنیم

تابع پله‌ای: این تابع با حد آستانه‌ای  $\theta$  است و بصورت زیر می‌باشد:

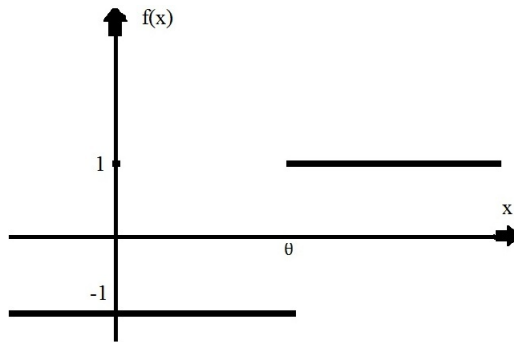
$$f(x) = \begin{cases} 1 & x \geq \theta, \\ 0 & x < \theta. \end{cases}$$



شکل ۵.۱: تابع پله‌ای

تابع پله‌ای دوقطبی: این تابع با حد آستانه‌ای  $\theta$  است و بصورت زیر تعریف می‌شود

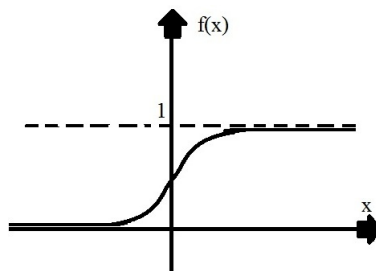
$$f(x) = \begin{cases} 1 & x \geq \theta, \\ -1 & x < \theta. \end{cases}$$



شکل ۶.۱: تابع پله‌ای دوقطبی

تابع سیگموئید: تابع سیگموئید<sup>۱</sup>، تابعی یکنوا، کراندار و مشتق پذیر است و می‌توان مشتق تابع را به کمک خود تابع بیان کرد و به شکل زیر است:

$$f(x) = \frac{1}{1 + e^{-x}}, \quad f'(x) = f(x)(1 - f(x)).$$

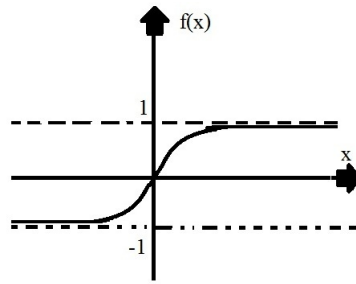


شکل ۷.۱: تابع سیگموئید

تابع سیگموئید قطبی: این تابع همان خواص تابع سیگموئید را دارد و بصورت زیر تعریف می‌شود:

$$g(x) = 2f(x) - 1 = 2 \left( \frac{1}{1 + e^{-x}} \right) - 1 = \frac{1 - e^{-x}}{1 + e^{-x}}, \quad g'(x) = \frac{1}{4}(1 + g(x))(1 - g(x)).$$

<sup>۱</sup> Sigmoid



شکل ۸.۱: تابع سیگموئید قطبی

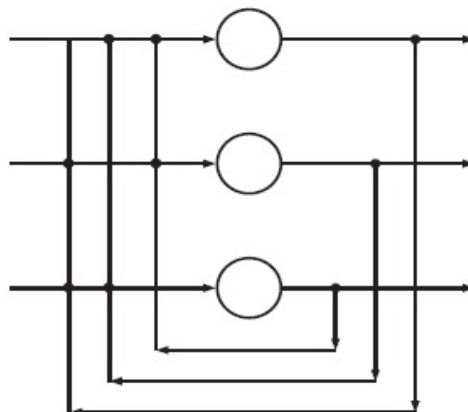
تابع همانی: این تابع بیشتر برای نورون‌های ورودی مورد استفاده قرار می‌گیرد و بصورت زیر بیان می‌شود:

$$f(x) = x.$$

چند تابع فعال سازی مهم دیگر را در جدول (۱.۱) آورده‌ایم.

### ۱.۲.۱.۱ شبکه‌های عصبی بازگشتی

در شبکه‌های عصبی بازگشتی<sup>۱۱</sup>، حداقل یک سیگنال برگشتی از یک نرون به همان نرون یا نرون‌های همان لایه و یا لایه قبل وجود دارد. چنین شبکه‌ای حافظه‌ای را نگه می‌دارد و حالت بعد نه تنها به سیگنال‌های ورودی بلکه به حالات قبل شبکه نیز وابسته است. این شبکه‌های عصبی معمولاً در مدل ریاضی به صورت معادلات دیفرانسیل مرتبه اول ظاهر می‌شوند.

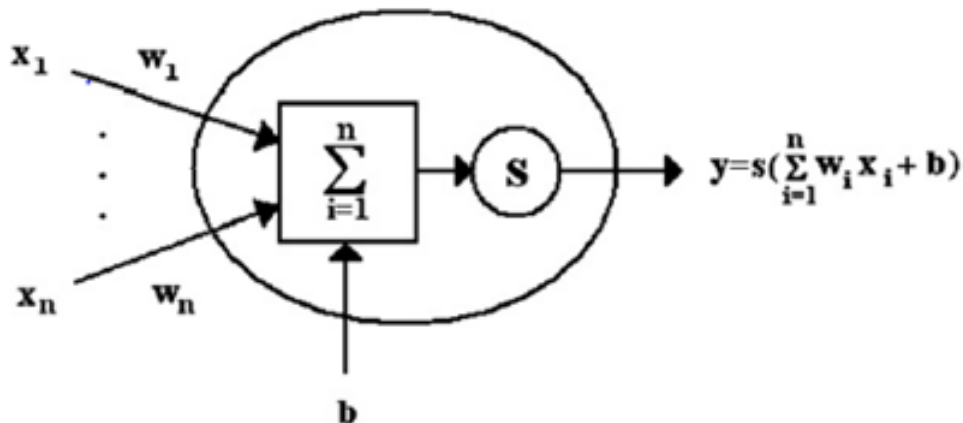


شکل ۹.۱: شبکه‌های عصبی دو لایه بازگشتی.

<sup>۱۱</sup>Feedback (Recurrent) Neural Network

## ۳.۱.۱ شبکه‌های عصبی پیش‌خورد

شبکه‌های عصبی مصنوعی<sup>۱۲</sup> یک گرایش هیجان‌انگیز از هوش مصنوعی است که شبیه به فرآیند یادگیری مغز انسان به منظور استخراج الگوهای از داده‌های تاریخی عمل می‌کند [۵۱]. برای سال‌های بسیاری این تکنولوژی بطور موفقیت آمیز برای گونه‌های گسترده‌ای از کاربردهای دنیای واقعی بکارگرفته شده است [۴۹]. پرسپترون<sup>۱۳</sup> برای اولین بار توسط روزنبلات<sup>۱۴</sup> در سال ۱۹۶۲ معرفی شد [۴۳] (شکل (۱۰.۱) را ببینید). پرسپترون ساده به یک یاددهنده احتیاج دارد که به شبکه بگوید که چه خروجی مطلوبی باید داشته باشد. آن‌ها شبکه‌های تحت نظارت هستند. دریک شبکه بدون نظارت، شبکه سازگار با پاسخ به ورودی آن است. شکل (۱۱.۱) مثالی از پرسپترون سه لایه را نشان می‌دهد. پرسپترون چندلایه با لایه‌های بیشتر از سه‌تا، از لایه‌های پنهان شده استفاده می‌کنند [۶۳، ۵۴].



شکل ۱۰.۱: ساختار شبکه عصبی مصنوعی (پرسپترون)

در پرسپترون‌های چندلایه واحدهای ورودی به واحدهای خروجی توسط یک نگاشت غیرخطی بهم متصل می‌شوند [۳۴]. مهمترین کاربرد پرسپترون چندلایه توانایی آنها در تقریب تابع است. از قضیه وجودی کولوموگروف<sup>۱۵</sup> می‌دانیم که یک پرسپترون سه لایه با  $n(2n+1)$  گره می‌تواند از هر تابع پیوسته  $n$  متغیر محاسبه شود [۴۱، ۵۲]. دقت تقریب به تعداد نرون‌ها<sup>۱۶</sup> در لایه‌های پنهان شده بستگی دارد و به تعداد لایه‌های پنهان شده بستگی ندارد [۱۲]. شبکه‌های عصبی چندلایه پیش‌خورد<sup>۱۷</sup> قابل یادگیری هستند. در برخی موارد مسأله‌ی آموزشی پیش‌خورد بدحالت است [۳۵].

شبکه‌های عصبی یک شکل از سیستم کامپیوتر چندپردازنده با عناصر ساده پردازش، ارتباطات درجه بالاتر، مقیاس پیام ساده<sup>۱۸</sup> و عکس‌العمل تطبیقی بین عناصر هستند. یک نرون بیولوژیکی<sup>۱۹</sup> ممکن است  $10^4$  ورودی متفاوت داشته باشد و ممکن است خروجی آن را به بسیاری از نرون‌های دیگر ارسال کند. شبکه عصبی مصنوعی هرچیزی را که با روش‌های محاسبات سنتی نمی‌توان انجام داد، می‌تواند

<sup>۱۲</sup>Artificial Neural Networks

<sup>۱۳</sup>Perceptron

<sup>۱۴</sup>Rosenblatt

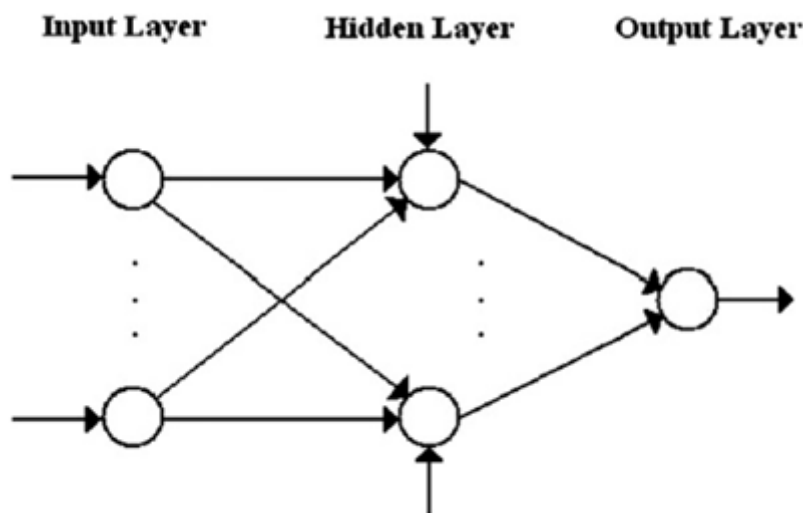
<sup>۱۵</sup>Existence Colomogorov Theorem

<sup>۱۶</sup>Neuron

<sup>۱۷</sup>FeedForward

<sup>۱۸</sup>Simple Message Scale

<sup>۱۹</sup> Biological Neural Network



شکل ۱۱.۱: پرسپترون چندلایه

انجام دهد. آنها می‌توانند برخی از کارهای محاسباتی که بسیار مشکل است را انجام دهند. از شبکه‌های عصبی پیش‌خورد بصورت یک مدل (تابع با متغیرهای زیاد) از داده‌های آموزشی آنها (داده‌های ورودی) به تنهایی استفاده می‌کنیم.

#### ۴.۱.۱ تاریخچه تکامل شبکه عصبی

در سال ۱۹۴۳ یک نرولوژیست به نام وارن مک کلاچ<sup>۲۰</sup> به همراه یک متخصص آمار به نام والتر پیتز<sup>۲۱</sup> اولین مدل ریاضی نرون را ارائه دادند [۶۴]. این نرون ساده یک عنصر محاسبه‌گر بود که عناصری به نام ورودی را در مقادیر ثابتی به نام وزن ضرب می‌کرد و آنها را از یک عملگر خطی عبور می‌داد و حاصل خروجی نرون را تشکیل می‌داد. در این مدل وزن‌ها مقادیر ثابتی بودند. در سال ۱۹۴۹ دونالد هب<sup>۲۲</sup> [۲۳] کتابی درباره نحوه یادگیری در مغز انسان نوشت. بر این مبنا قانون یادگیری برای نرون را ارائه داد که به قانون هب معروف است و بعدها در تعلیم شبکه‌های عصبی از آن استفاده شد. در سال ۱۹۵۴ فارلی و کلارک<sup>۲۳</sup> [۱] مدلی از نرون‌ها را به‌طور تصادفی به هم وصل کردند و قانون هب را پیاده‌سازی نمودند و نشان دادند که می‌توان دو الگوی ورودی را با این قانون تعلیم از هم تشخیص داد. با پیشرفت علم کامپیوتر اولین مدل از نرون عصبی مصنوعی توسط ناتانیال روچستر<sup>۲۴</sup> محقق شرکت IBM در سال ۱۹۵۸ با استفاده از کامپیوتر شبیه‌سازی شد. روزنبلات [۲۷، ۲۸] در سال ۱۹۵۸ نرون ساده مک‌کلاچ پیتز را اصلاح کرد و به آن قابلیت یادگیری و سازگاری را اضافه کرد و این نرون را پرسپترون نامید. قانون تعلیم پرسپترون<sup>۲۵</sup> اولین قانون رسمی برای شبکه‌های عصبی است.

<sup>۲۰</sup>Warren McCulloch

<sup>۲۱</sup>Walter Pitts

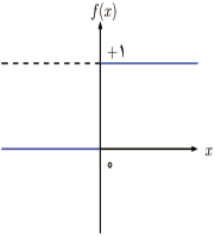
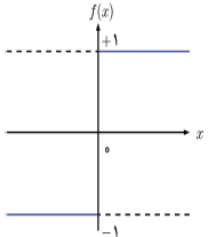
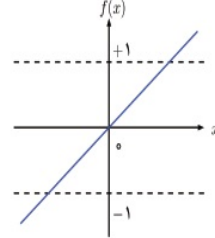
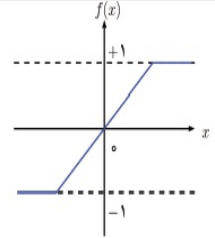
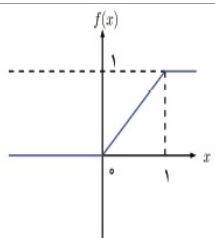
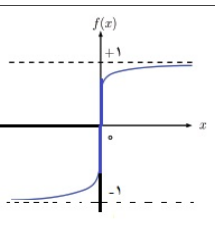
<sup>۲۲</sup>Donald Hebb

<sup>۲۳</sup>Farley and Clark

<sup>۲۴</sup>Nathanial Rochester

<sup>۲۵</sup>Perceptron

جدول ۱.۱: توابع فعال سازی

شکل تابع	تعریف تابع	نام	ردیف
	$f(x) = \begin{cases} 0 & x < 0, \\ 1 & x \geq 0. \end{cases}$	آستانه‌ای دو مقداره	۱
	$f(x) = \begin{cases} -1 & x < 0, \\ 1 & x \geq 0. \end{cases}$	آستانه‌ای دو مقداره متقارن	۲
	$f(x) = x$	خطی	۳
	$f(x) = \begin{cases} -1, & x < -1, \\ x, & -1 \leq x \leq 1, \\ 1, & x > 1. \end{cases}$	آستانه‌ای خطی متقارن	۴
	$f(x) = \begin{cases} 0, & x < 0, \\ x & 0 \leq x \leq 1, \\ 1, & x > 1. \end{cases}$	آستانه‌ای خطی	۵
	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	تانژانت هایپربولیک	۶

برنارد ویدرو<sup>۲۶</sup> و مارسین هاف<sup>۲۷</sup> و همکارانشان یک مدل سه سطحی از نرون‌ها را ابداع کردند که آدلاین<sup>۲۸</sup> نام‌گذاری شد و مختصر شده حروف "عصر خطی تطبیقی"<sup>۲۹</sup> بود [۱۶، ۱۷، ۱۸، ۱۹]. و برای آدلاین یک قانون تعلیم بر مبنای مشتق‌گیری بیان نمودند. این قانون تعلیم به  $\alpha - LMS$  معروف است. از متصل کردن چند آدلاین به یکدیگر مدلین<sup>۳۰</sup> ساخته شد. ویدرو و هاف از شبکه عصبی مدلین برای حذف اکو از خطوط تلفن استفاده نمودند.

سال ۱۹۶۹ آغاز یک دوره رکود برای شبکه عصبی بود. پاپرت و مینسکی<sup>۳۱</sup> [۴۵] کتابی به نام پرسپترون نوشتند و نشان دادند که اگرچه گیت‌های AND و OR را می‌توان با استفاده از پرسپترون پیاده سازی کرد ولی گیت XOR قابل پیاده سازی نمی‌باشد و لذا نمی‌توان از شبکه عصبی برای پردازش اطلاعات استفاده کرد. الگوریتم پس انتشار<sup>۳۲</sup> پال و ریاس<sup>۳۳</sup> [۴۸] در سال ۱۹۷۴ ارائه شد و بعدها توسط رامل هارت<sup>۳۴</sup> [۲۴] به‌طور مستقل کشف گردید. این الگوریتم از زمان پیدایش به‌طور گسترده به‌عنوان یک الگوریتم آموزش در شبکه‌های عصبی پیش‌خورد<sup>۳۵</sup> مورد استفاده قرار گرفته است. سال ۱۹۸۲ تا ۱۹۸۶ را می‌توان تولد دوباره شبکه عصبی دانست. در این سال‌ها دو اتفاق مهم در این زمینه رخ داد. اول ارائه شبکه عصبی بازگشتی و مفهوم تابع انرژی توسط جان هاپفیلد<sup>۳۶</sup> [۳۶] بود که در سال ۱۹۸۲ منتشر شد. دومین اتفاق، حل مسأله فروشنده دوره‌گرد (TSP) توسط شبکه هاپفیلد بود [۳۷]. هاپفیلد نشان داد که از این شبکه بازگشتی می‌توان برای حل مسائل بهینه‌سازی استفاده کرد و افق جدیدی را در شبکه‌های عصبی مصنوعی گشود.

## ۵.۱.۱ تاریخچه حل مسائل بهینه‌سازی با استفاده از شبکه‌های عصبی

طیف گسترده‌ای از مسائل مهم در رشته‌های علوم پایه و مهندسی از جمله کنترل بهینه، طراحی ساختمان، پردازش تصویر، تقریب توابع، تحلیل رگرسیون و غیره قابل تبدیل به مسائل بهینه‌سازی خطی و غیرخطی هستند. در دهه‌های گذشته الگوریتم‌های بهینه‌سازی عددی فراوانی برای حل مسائل بهینه‌سازی خطی و غیرخطی ارائه شده‌اند. از آن‌جا که زمان مورد نیاز برای حل مسائل بهینه‌سازی و به‌خصوص مسائل بهینه‌سازی غیرخطی بسیار وابسته به بعد و ساختار مسأله است بنابراین الگوریتم‌های عددی کارایی کمی از خود نشان می‌دهند. یک رهیافت امیدوارکننده برای حل مسائل بهینه‌سازی با ابعاد بالا استفاده از شبکه‌های عصبی مصنوعی می‌باشد.

در سال ۱۹۸۶ تانک و هاپفیلد [۲۲] اولین شبکه عصبی را برای حل مسائل برنامه‌ریزی خطی معرفی کردند و نشان دادند که وضعیت شبکه در هر تکرار به‌گونه‌ای تغییر می‌کند که تابع انرژی متناظر با آن به‌طور یکنواخت کاهش می‌یابد تا جایی که به نقطه مینیمم خود می‌رسد و این نقطه مینیمم متناظر با نقطه تعادل شبکه عصبی می‌باشد. آن‌ها این شبکه عصبی را توسط یک مدار الکتریکی پیاده سازی

<sup>۲۶</sup>Bernard Widrow

<sup>۲۷</sup>Marcian Hoff

<sup>۲۸</sup>ADELIN

<sup>۲۹</sup>Adaptive Linear Elements

<sup>۳۰</sup>MADALINE

<sup>۳۱</sup>Papert and Minsky

<sup>۳۲</sup>Back Propagation

<sup>۳۳</sup>Paul Werbos

<sup>۳۴</sup>Rumelhart

<sup>۳۵</sup>Feed Forward

<sup>۳۶</sup>John Hopfield

کردند. هم‌چنین از این شبکه برای حل مسأله فروشنده دوره‌گرد با ۳۰ شهر استفاده نمودند. این شبکه دارای نقص‌هایی بود به‌خصوص این‌که نقطه تعادل شبکه در شرایط KKT<sup>۳۷</sup> صدق نمی‌کرد و لذا جواب مطلوبی از مسأله حاصل نمی‌شد. با این وجود کارهای هاپفیلد انگیزه بسیار خوبی را برای محققین به‌وجود آورد تا در این زمینه فعالیت کنند. کندی و چا<sup>۳۸</sup> [۴۴] با افزودن یک پارامتر جریمه متناهی به شبکه هاپفیلد آن را توسعه دادند و از آن برای حل مسأله برنامه‌ریزی غیرخطی محدب استفاده نمودند. با پارامترهای جریمه این شبکه در یافتن نقطه بهینه دقیق ناتوان است به‌خصوص اگر پارامتر جریمه بزرگ باشد این شبکه به‌سختی عمل می‌کند.

برای جلوگیری از بکار بردن پارامتر جریمه یک شبکه عصبی سوئیچ-خازن<sup>۳۹</sup> توسط رودریگز-وازکز<sup>۴۰</sup> و همکاران [۱۴] معرفی شد. هم‌چنین ما و شانبلات<sup>۴۱</sup> [۲۱] یک شبکه عصبی دو فازی ارائه کردند که فاز اول شبکه مشابه شبکه کندی و چا بود و در فاز دوم آن مسیر شبکه به جواب دقیق مسأله همگرا می‌شد بنابراین این روش جواب‌های دقیق‌تری نسبت به شبکه کندی و چا ارائه می‌داد. مشکل این شبکه در این بود که پایداری فاز دوم شبکه بستگی به انتخاب یک مقدار بزرگ از پارامتر جریمه داشت و لذا اگرچه تأثیر پارامتر جریمه در این روش کاهش یافته بود و جواب‌های حاصل دقیق‌تر بودند اما هنوز مستقل از پارامتر نبود.

ژانگ و کنستانتینیدس<sup>۴۲</sup> [۶۰] بر مبنای روش لاگرانژ یک شبکه عصبی ارائه کردند که کاملاً مستقل از پارامتر جریمه و قادر به حل مسائل غیرخطی بود. نقطه تعادل این شبکه عصبی در شرایط بهینگی مرتبه اول و دوم صدق می‌کرد و هم‌چنین شبکه حاصل همگرا بود. در سال ۱۹۹۳ بوزردوم و پتیسین<sup>۴۳</sup> [۱۰] شبکه‌ای را بر مبنای گرادیان و روش تصویر و مستقل از پارامتر جریمه ابداع کردند که تنها قادر به حل مسائل درجه دوم با متغیرهای کران‌دار بود. این روش در عمل روش کارایی بود اما نمی‌توانست مسائل کلی برنامه‌ریزی خطی و درجه دوم را حل کند.

زیا<sup>۴۴</sup> و همکاران [۶۷، ۶۶، ۶۸، ۶۹] چندین مدل را برای حل مسائل بهینه‌سازی غیرخطی با قیود خطی و غیرخطی ارائه کردند. هم‌چنین چندین مدل شبکه عصبی برای حل مسائل بهینه‌سازی غیرخطی توسط عفتی و همکاران [۴۶، ۲، ۴۷، ۵۵، ۵۷، ۵۶] ارائه شده است.

## ۶.۱.۱ شبکه عصبی (نرون مک کلاچ) برای تابع AND

در این نوع شبکه عصبی هرگاه دو مقدار دریافتی درست باشند تابع AND پاسخ درست و هرگاه یکی از دو مقدار دریافتی نادرست باشند تابع پاسخ نادرست را برمی‌گرداند.

$$y = f(1 \times x_1 + 1 \times x_2) = f(x_1 + x_2) = \begin{cases} 1 & x_1 + x_2 \geq 2, \\ 0 & x_1 + x_2 < 2. \end{cases}$$

در اینجا حد آستانه‌ای  $\theta = 2$  می‌باشد.

<sup>۳۷</sup>Karush-Kuhn-Tucker

<sup>۳۸</sup>Kennedy and Chua

<sup>۳۹</sup>Switched-Capacitor

<sup>۴۰</sup>Rodriguez-Vazquez

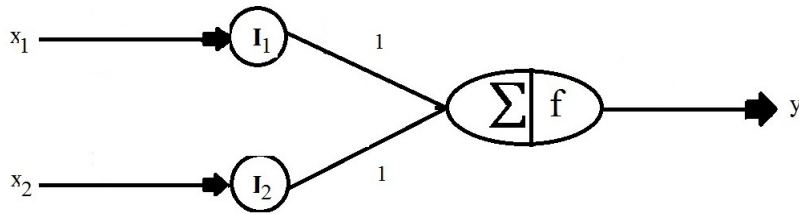
<sup>۴۱</sup>Maa and Shanblatt

<sup>۴۲</sup>Zhang and Constantinides

<sup>۴۳</sup>Bouzerdoum and Pattison

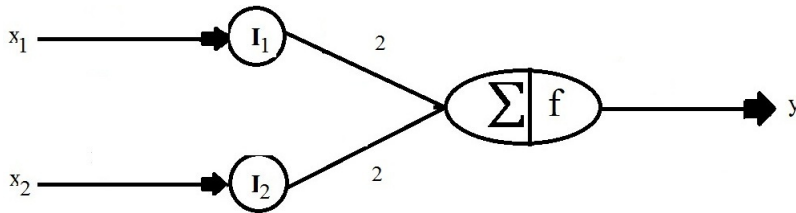
<sup>۴۴</sup>xia





شکل ۱۲.۱: نرون مک کلاچ برای تابع AND

### ۷.۱.۱ شبکه عصبی (نرون مک کلاچ) برای تابع OR



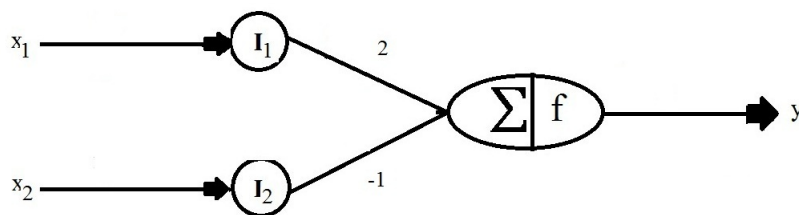
شکل ۱۳.۱: نرون مک کلاچ برای تابع OR

در این نوع شبکه عصبی هرگاه یکی از دو مقدار دریافتی درست باشند تابع OR پاسخ درست و هرگاه هر دو دو مقدار دریافتی نادرست باشند تابع پاسخ نادرست را برمی‌گرداند.

$$y = f(2 \times x_1 + 2 \times x_2) = f(2x_1 + 2x_2) = \begin{cases} 1 & 2x_1 + 2x_2 \geq 2, \\ 0 & 2x_1 + 2x_2 < 2. \end{cases}$$

در اینجا حد آستانه‌ای  $\theta = 2$  می‌باشد.

### ۸.۱.۱ شبکه عصبی (نرون مک کلاچ) برای تابع AND NOT



شکل ۱۴.۱: نرون مک کلاچ برای تابع AND NOT

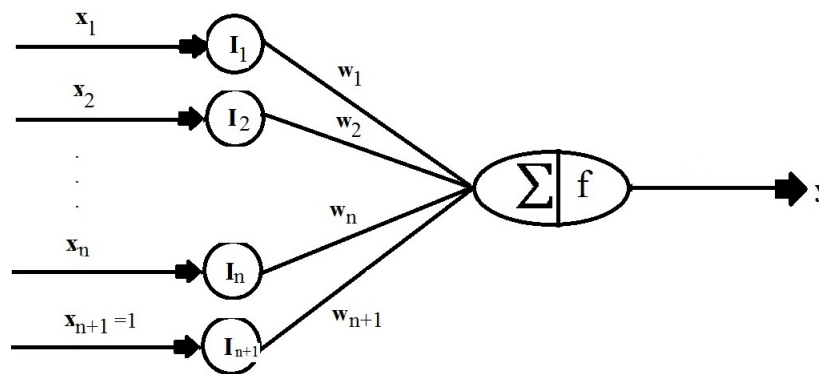
در این نوع شبکه عصبی هرگاه مولفه‌ی اول درست و مولفه‌ی دوم نادرست باشند تابع AND NOT پاسخ درست و در غیر اینصورت پاسخ نادرست می‌دهد.

$$y = f(2 \times x_1 + (-1) \times x_2) = f(2x_1 - x_2) = \begin{cases} 1 & 2x_1 - x_2 \geq 2, \\ 0 & 2x_1 - x_2 < 2. \end{cases}$$

در اینجا حد آستانه‌ای  $\theta = 2$  می‌باشد.

### ۹.۱.۱ شبکه عصبی هب

در ادامه الگوریتم آموزش را برای یک شبکه با  $n$  ورودی و یک خروجی متشکل از دولا به ارائه داده شده است.



شکل ۱۵.۱: شبکه عصبی هب

### الگوریتم هب

گام اول: مقدار دهی اولیه به تمام وزن‌ها  $w_i = 0, i = 1, 2, \dots, n+1$  همان بایاس است که ورودی ۱ دارد.

گام دوم: مولفه‌های بردار ورودی  $x_i (i = 1, 2, \dots, n+1)$  را روی واحدهای ورودی تنظیم <sup>۴۵</sup> می‌کنیم.

گام سوم: خروجی مورد انتظار مربوط به بردار ورودی گام قبل را روی واحد خروجی تنظیم می‌کنیم.

گام چهارم: وزن‌های موجود روی یال‌ها <sup>۴۶</sup> را بصورت زیر بروز می‌کنیم.

$$w_i(new) = w_i(old) + x_i y, \quad i = 1, 2, \dots, n+1.$$

<sup>۴۵</sup>Set

<sup>۴۶</sup>Edge

مثال ۲.۱۰.۱. الگوریتم هب را برای تابع AND با ورودی‌های دوقطبی و خروجی دوقطبی تکرار می‌کنیم. بنابراین داده‌های آموزشی بصورت زیر خواهند بود:

$$\begin{cases} (x_1, x_2, 1) & y \\ (1, 1, 1) & 1 \\ (1, -1, 1) & -1 \\ (-1, 1, 1) & -1 \\ (-1, -1, 1) & -1 \end{cases}$$

$$w_1 = w_2 = w_3 = 0$$

که در شکل‌های (۱۶.۱) و (۱۷.۱) روند الگوریتم نمایش داده شده است.

### ۱۰.۱.۱ شبکه عصبی آدالین

این شبکه چند ورودی دارد که یکی از ورودی‌ها همواره مقدار یک را به شبکه ارسال می‌کند (بایاس). نرون آدالین سیگنال‌ها را پس از ترکیب با وزن‌های روی یال‌ها جمع می‌کند. تابع فعال سازی روی نرون آدالین همان تابع همانی است، بنابراین همین مجموع به عنوان خروجی به بیرون فرستاده می‌شود و اختلاف بین مقدار خروجی و مقدار مورد انتظار مبنای الگوریتم آموزش محسوب می‌شود.

### الگوریتم آدالین

گام اول: به وزن‌ها مقادیر تصادفی بین صفر و یک نسبت دهید و ضریب آموزشی  $\alpha$  را عددی در بازه  $[0, 1]$  در نظر بگیرید.

گام دوم: تا زمانی که شرط توقف نادرست باشد گام‌های سوم تا هفتم را تکرار کنید.

گام سوم: برای هر جفت داده  $((x_1, x_2, \dots, x_n), y)$  گام‌های چهارم تا ششم را اجرا کنید.

گام چهارم: مولفه‌های بردار ورودی  $x_i$  ( $i = 1, 2, \dots, n+1$ ) را روی واحدهای ورودی تنظیم می‌کنیم.

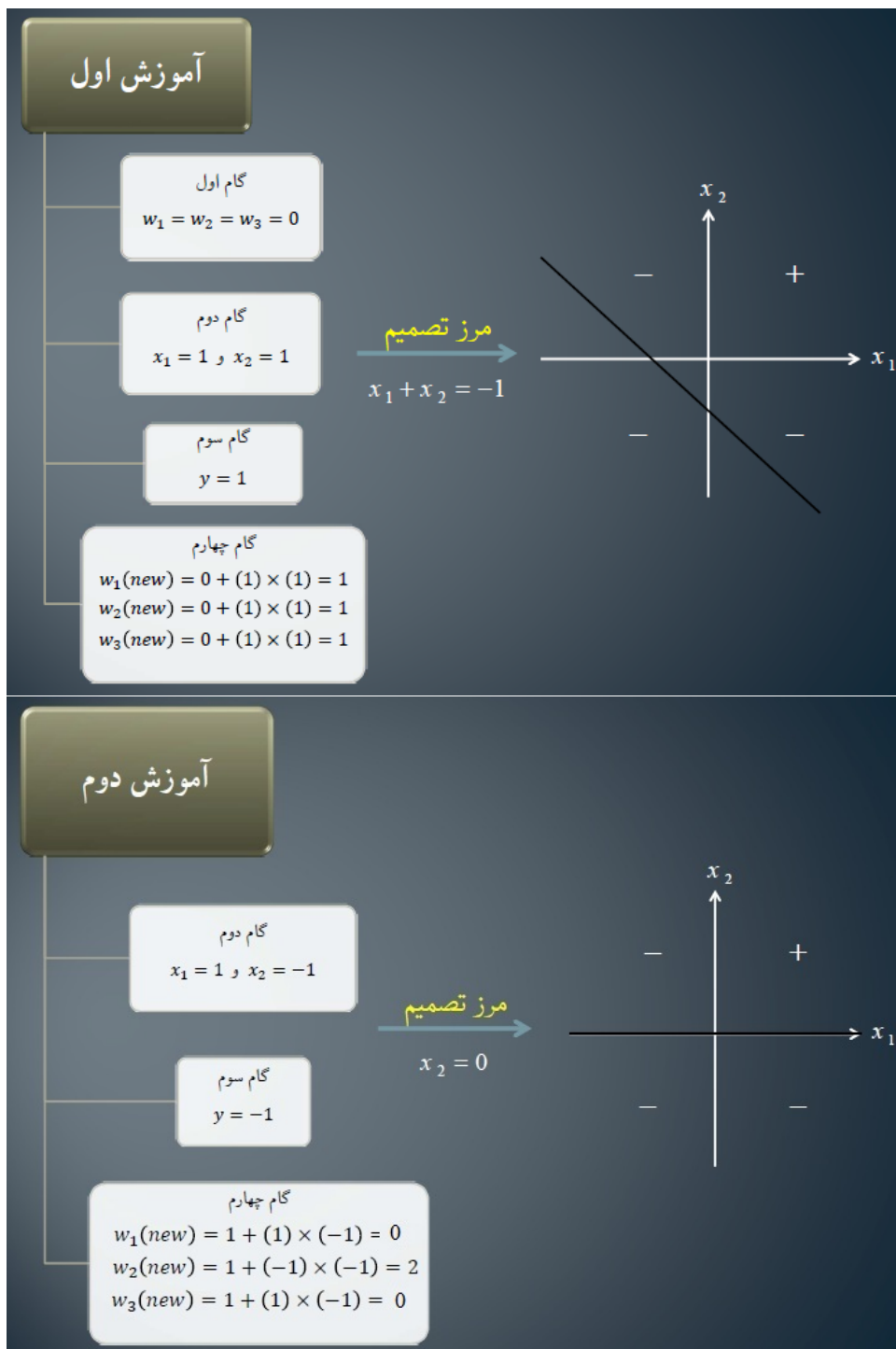
گام پنجم:

$$net = \sum_{i=1}^n w_i x_i,$$

$$I(net) = net.$$

گام ششم: تنظیم وزن‌ها

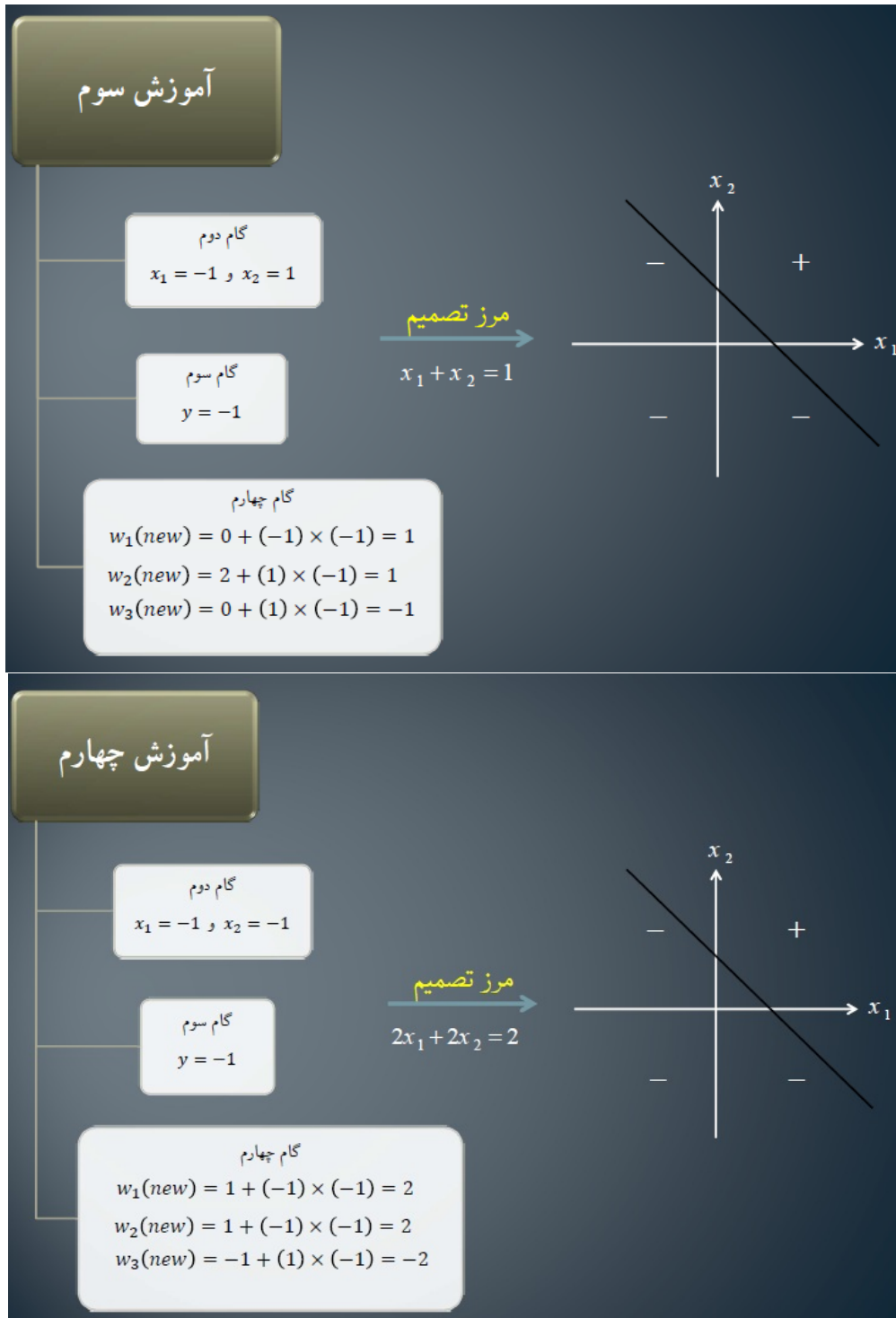
$$w_i(new) = w_i(old) + \alpha(y - net), \quad i = 1, 2, \dots, n+1$$



شکل ۱۶.۱: الگوریتم هب برای مثال (۲.۱.۱)

**گام هفتم:** اگر بیشترین تغییرات انجام گرفته برای وزن‌ها در گام دوم کمتر از یک تلورانس معین باشد، متوقف شوید در غیر اینصورت ادامه دهید.

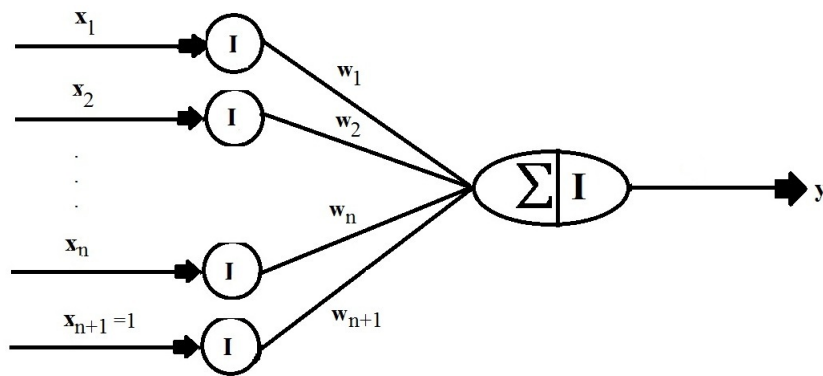
**مثال ۳.۱.۱.** در این مثال یک آدالین را برای تابع AND با ورودی‌های دودویی و خروجی دوقطبی را در نظر بگیرید. همچنین ضریب آموزشی  $\alpha = 0.1$  و تلورانس خطا را برابر یک می‌گیریم. وزن‌های



شکل ۱۷.۱: الگوریتم هب برای مثال (۲.۱.۱)

حاصله بصورت زیر خواهند بود:

$$w_1 = 1, \quad w_2 = 2, \quad w_3 = \frac{-3}{2}$$



شکل ۱۸.۱: شبکه عصبی آدالاین

و مرز تصمیم بصورت زیر می‌باشد

$$x_1 + 2x_2 - \frac{3}{4} = 0.$$

مثال ۴.۱.۱. در این مثال یک آدالاین را برای تابع AND با ورودی‌های دودویی و خروجی دوقطبی را در نظر بگیرید. همچنین ضریب آموزشی  $\alpha = 0.1$  و تلورانس خطا را برابر یک می‌گیریم. وزن‌های حاصله بصورت زیر خواهند بود:

$$w_1 = \frac{1}{4}, \quad w_2 = \frac{1}{4}, \quad w_3 = \frac{-1}{4}$$

و مرز تصمیم بصورت زیر می‌باشد

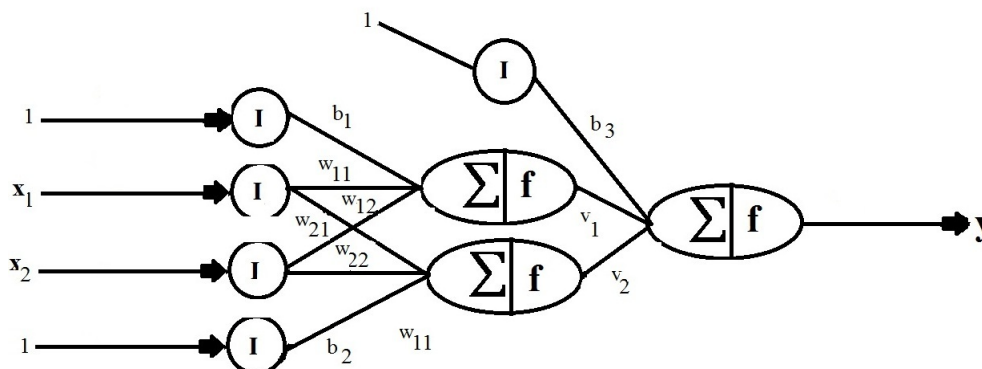
$$\frac{1}{4}x_1 + \frac{1}{4}x_2 - \frac{1}{4} = 0.$$

### ۱۱.۱.۱ شبکه عصبی مادالاین

یک مادالاین، ترکیبی از چند آدالاین می‌باشد که در یک شبکه چندلایه مرتب شده‌اند. در اینجا ماروی یک مادالاین با یک لایه مخفی (مربک از دو آدالاین) و یک خروجی متمرکز می‌شویم. در زیر یک مادالاین با دو ورودی و یک خروجی نشان داده شده است. این شبکه شامل دو آدالاین با ورودی‌های  $x_1, x_2$  و خروجی‌های  $z_1, z_2$  می‌باشد. و توابع بکاربرده شده در آدالاین‌ها توابع آستانه‌ای دو مقداری متقارن می‌باشد و خروجی  $y$  تابعی غیر خطی از بردار ورودی  $(x_1, x_2)$  می‌باشد.

### الگوریتم مادالاین

در این الگوریتم فقط وزن‌های لایه اول تنظیم می‌شوند و وزن‌های لایه بعدی در مدت آموزش ثابت می‌مانند. وزن‌ها به گونه‌ای هستند که اگر  $z_1$  یا  $z_2$  یا هر دو سیگنال ۱ بفرستند،  $y$  سیگنال ۱ و اگر هر دو



شکل ۱۹.۱: شبکه عصبی مادالاین

سیگنال ۱ - بفرستند  $y$  سیگنال ۱ - را تولید می‌کند بنابراین داریم:

$$v_1 = \frac{1}{4}, \quad v_2 = \frac{1}{4}, \quad b_3 = \frac{1}{4}$$

گام اول:  $v_1 = v_2 = b_3 = \frac{1}{4}$  و به وزن‌ها مقادیر تصادفی بین صفر و یک نسبت دهید و ضریب آموزشی  $\alpha$  را عددی در بازه  $[0, 1]$  در نظر بگیرید.

گام دوم: تا زمانی که شرط توقف نادرست باشد گام‌های سوم تا نهم را تکرار کنید.

گام سوم: برای هر جفت داده  $((x_1, x_2), y)$  گام‌های چهارم تا هشتم را اجرا کنید.

گام چهارم: بردار ورودی  $(x_1, x_2)$  را روی واحدهای ورودی تنظیم می‌کنیم.

گام پنجم: ورودی‌های مربوط به آدالین‌ها را محاسبه کنید.

$$net_1 = b_1 + x_1 w_{11} + x_2 w_{21},$$

$$net_2 = b_2 + x_1 w_{12} + x_2 w_{22}.$$

گام ششم: محاسبه خروجی هر آدالین

$$z_1 = f(net_1),$$

$$z_2 = f(net_2).$$

گام هفتم: تعیین خروجی شبکه

$$net = b_3 + z_1 v_1 + z_2 v_2, \quad f(net)$$

گام هشتم: اگر  $y = f(net)$  و وزن‌ها تغییر نخواهند کرد در غیر این صورت:

$$\text{if } y = 1, \quad w_{ij}(new) = w_{ij}(old) + \alpha(1 - net_j)x_j, \quad b_j(new) = b_j(old) + \alpha(1 - net_j),$$

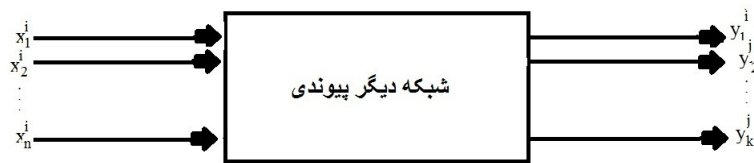
$$\text{if } y = -1, \quad w_{ij}(new) = w_{ij}(old) + \alpha(-1 - net_j)x_j, \quad b_j(new) = b_j(old) + \alpha(-1 - net_j).$$

گام نهم: اگر تغییرات در وزن‌ها متوقف شود یا به یک سطح قابل قبول برسیم، متوقف شوید.

### ۱۲.۱.۱ شبکه‌های حافظه پیوندی

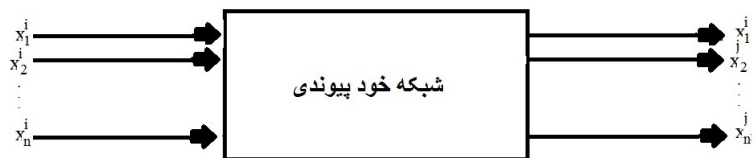
شبکه‌های حافظه پیوندی شبکه‌هایی تک لایه می‌باشند و وزن‌های موجود در شبکه، عمل ذخیره سازی الگوهای پیوندی را بر عهده دارند. این نوع شبکه عصبی بطور کلی به سه دسته تقسیم میشوند:

- شبکه‌های عصبی دیگر پیوندی:
- شبکه‌های عصبی دیگر پیوندی  $p$  بردار  $(x^1, x^2, \dots, x^n)$  در فضای  $n$  بعدی را به  $p$  بردار خروجی  $k$  بعدی  $(y^1, y^2, \dots, y^k)$  می‌برد.



شکل ۲۰.۱: شبکه عصبی دیگر پیوندی

- شبکه‌های عصبی خود پیوندی:
- شبکه‌های عصبی خود پیوندی حالت خاصی از شبکه‌های عصبی دیگر پیوندی است که هر بردار را با خودش مرتبط می‌سازد.



شکل ۲۱.۱: شبکه عصبی خود پیوندی

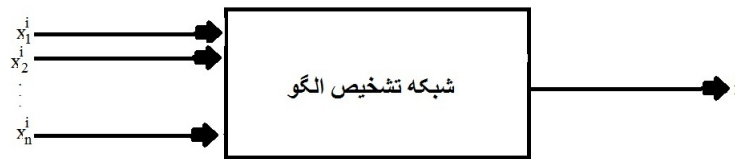
- شبکه‌های تشخیص الگو:
- در شبکه تشخیص الگو هر بردار  $x^i$  از فضای  $n$  با اسکالر  $i$  پیوند داده می‌شود.

### ۱۳.۱.۱ شبکه‌های عصبی هاپفیلد

#### ۱.۱۳.۱.۱ شبکه عصبی مصنوعی هاپفیلد

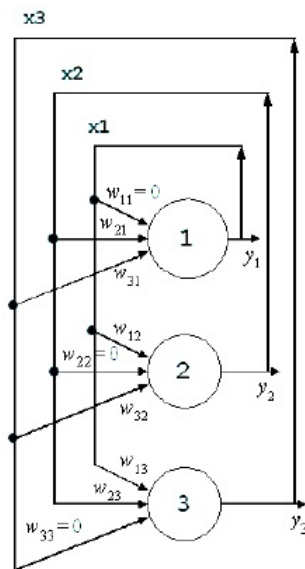
شبکه هاپفیلد یک شبکه بازگشتی است. به این مفهوم که در آن خروجی هر نرون به ورودی تمام نرون‌های دیگر متصل است. پس از اعمال ورودی اولیه به شبکه، خروجی آن محاسبه می‌شود و در قدم بعد به





شکل ۲۲.۱: شبکه عصبی تشخیص الگو

عنوان ورودی به شبکه اعمال می‌شود. این فرآیند آنقدر تکرار می‌شود تا خروجی شبکه به یک نقطه ثابت همگرا شود. در اینجا شبکه به عنوان یک سیستم دینامیکی که وضعیتش در هر تکرار تغییر می‌کند، عمل می‌کند. و همگرایی شبکه یعنی رسیدن پاسخ شبکه به جوابی که بر اثر تکرار تغییر نکند، در این صورت می‌توان گفت که شبکه به حالت پایدار یا تعادل خود رسیده است. شکل (۲۳.۱) یک شبکه هاپفیلد با سه نرون را نشان می‌دهد [۹]. شبکه هاپفیلد به دو نوع گسسته و



شکل ۲۳.۱: ساختار شبکه هاپفیلد با سه نرون

پیوسته تقسیم می‌شود. شبکه هاپفیلد گسسته یک شبکه بازگشتی شامل نرون‌هایی است که تابع فعالیت آن‌ها یک تابع گسسته است. در واقع هر نرون در شبکه گسسته یک مدل مکلوث<sup>۴۷</sup> و پیتس<sup>۴۸</sup> است [۶۴]. اما در شبکه هاپفیلد پیوسته که بحث اصلی این بخش است، هر نرون دارای یک تابع فعالیت به طور پیوسته مشتق پذیر است.

<sup>۴۷</sup>Warren McCulloch

<sup>۴۸</sup>Walter Pitts

## ۲.۱۳.۱.۱ نحوه عملکرد شبکه هاپفیلد پیوسته

در سال ۱۹۸۲ [۳۶] هاپفیلد بر مبنای ایده تابع انرژی یا تابع لیاپانوف برای سیستم‌های دینامیکی شبکه عصبی بازگشتی را ارائه داد که به نام خود او شبکه عصبی هاپفیلد نام گذاری و اولین بار از آن برای شناسایی الگوهای دارای نویز استفاده شد. همانطور که در ادامه بحث، خواهد آمد، هر شبکه هاپفیلد پیوسته یک سیستم دینامیکی است که دارای یک تابع انرژی می‌باشد. هدف از اجرای یک شبکه هاپفیلد، تشخیص الگوهای ناشناس است. با در نظر گرفتن منحنی تابع انرژی شبکه درمی‌یابیم که منحنی دارای نقاط مینیمم است که می‌تواند به عنوان مکان ذخیره سازی الگوها یا جواب‌های مطلوب شبکه مورد استفاده قرار گیرد.

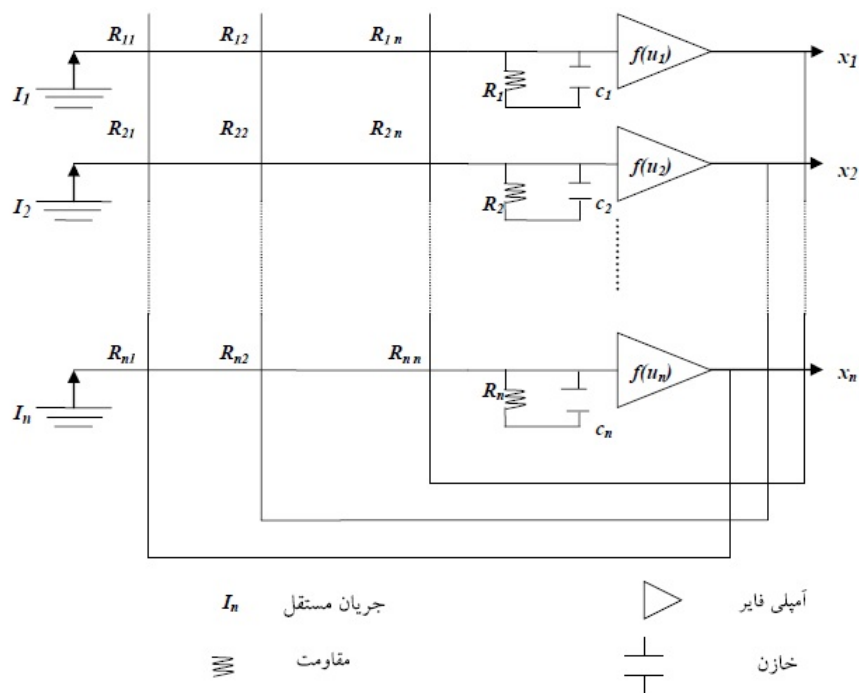
این حفره‌ها یا نقاط مینیمم محل نقاط جذب شبکه نامیده می‌شوند. نقاط جذب به بیان ریاضی همان نقاط ساکن پایدار مجانبی سیستم دینامیکی هستند و در شبکه هاپفیلد، نقاط جذب نقاطی هستند که شبکه به آن‌ها همگرا می‌شود.

توصیف هندسی مساله همگرایی را می‌توان به این صورت نیز بیان کرد که الگوهای ورودی که وضعیت اولیه شبکه را مشخص می‌کند، به سمت کوچکترین ناحیه‌ای که شامل یکی از حفره‌های جذب (نقاط ساکن پایدار مجانبی) می‌باشد، نزدیک می‌شود و هنگامی که به نزدیکی آن رسید از هر طرف که حرکت کند نقطه جذب آن را به سمت خود می‌کشاند و نهایتاً در آن ساکن می‌ماند. بخش بعد ما را در بهتر درک کردن ساختار شبکه عصبی پیوسته هاپفیلد و پایداری آن یاری می‌دهد.

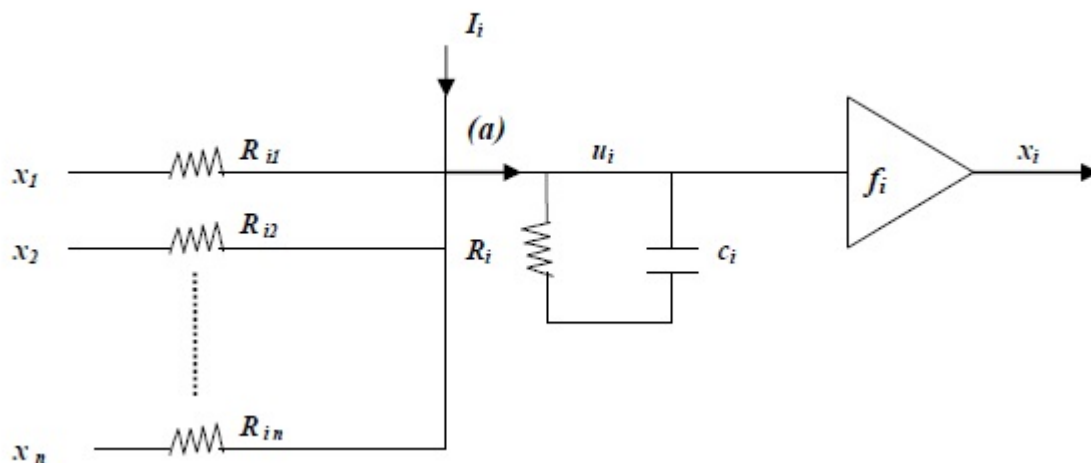
## ۳.۱۳.۱.۱ ساختار شبکه هاپفیلد پیوسته

هاپفیلد این شبکه را بر مبنای یک مدار الکتریکی ارائه می‌دهد. مدار الکتریکی در شکل (۲۴.۱) را در نظر بگیرید [۵۰]. هر آمپلی فایر ولتاژ ورودی  $u_i$  را به ولتاژ  $x_i = f(u_i)$  تبدیل می‌کند. تابع  $f$  را یک تابع مشتق پذیر و به طور یکنواخت صعودی مانند تابع سیگموئید  $\tanh(\beta u_i)$  قرار می‌دهیم. هر مقاومت  $R_{ij}$  ولتاژ خروجی  $x_j$  را به آمپلی فایر  $i$ -ام هدایت می‌کند. جریان  $I_i$  به عنوان اولین ورودی یا بایاس برای آمپلی فایر  $i$  در نظر گرفته می‌شود. مدار فوق به عنوان شبکه هاپفیلد شناخته می‌شود.

در این شبکه ولتاژ اولیه  $x(0)$  که همان ورودی ناشناس است به عنوان ورودی اولیه به شبکه اعمال می‌شود. ولتاژ خروجی  $x = [x_1, \dots, x_n]$  اندازه گیری می‌شود و سپس  $x_1$  مجدداً به عنوان ورودی وارد شبکه می‌شود. این فرآیند آنقدر تکرار می‌شود تا ولتاژ خروجی شبکه تغییر نکند یا در همسایگی نقطه‌ای ثابت قرار گیرد. در این صورت گوئیم شبکه به حالت پایدار خود رسیده است. لذا این فرآیند را می‌توان به صورت یک معادله بازگشتی در نظر گرفت. پس شبکه هاپفیلد یک شکل بازگشتی است و اثبات پایداری شبکه از روی ورودی معادله دینامیکی مدار آن به شکل زیر نتیجه می‌شود. شکل (۲۵.۱) یک واحد از این شبکه را به عنوان یک نرون شبکه نمایش می‌دهد. معادله دینامیکی که تغییر وضعیت  $x_i$  را توضیح



شکل ۲۴.۱: مدل الکتریکی شبکه هاپفیلد



شکل ۲۵.۱: یک واحد از نرون شبکه هاپفیلد

می‌دهد از قانون جریان کیرشهف گرفته شده. در گره (a) داریم:

$$\sum_{j=1}^n \frac{1}{R_{ij}} x_j - \sum_{j=1}^n \frac{1}{R_{ij}} u_i - \frac{u_i}{R_i} - C_i \frac{du_i}{dt} + I_i = 0, \quad (1.1)$$

$$C_i \frac{du_i}{dt} = \sum_{j=1}^n \frac{1}{R_{ij}} (x_j - u_i) - \frac{u_i}{R_i} + I_i, \quad (2.1)$$

$$C_i \frac{du_i}{dt} = -\alpha_i \times u_i + \sum_{j=1}^n x_{ij} f(u_j) + I_i. \quad (3.1)$$

که در آن  $w_{ij} = \frac{1}{R_{ij}}$ ،  $\alpha_i = \sum_{j=1}^n \frac{1}{R_{ij}} + \frac{1}{R_i}$  و  $x_i = f(u_i)$  معادله (۳.۱) به صورت برداری زیر نوشته می‌شود.

$$C \frac{dU}{dt} = -\alpha \times u + w \times x + \theta, \quad (4.1)$$

که در آن

$$C = (C_1, \dots, C_n), \quad \alpha = (\alpha_1, \dots, \alpha_n), \quad x = F(u) = (f(u_1), \dots, f(u_n))^T,$$

$$\theta = [I_1, \dots, I_n]^T, \quad w = \begin{bmatrix} w_{11} & \dots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \dots & w_{nn} \end{bmatrix}.$$

نقطه ساکن معادله دینامیکی (۴.۱) با قرار دادن  $\frac{du}{dt} = 0$  حاصل می‌شود. در این صورت:

$$\alpha \times u = w \times x + \theta.$$

معادله (۴.۱) سیستم دینامیکی شبکه عصبی هاپفیلد می‌باشد. هاپفیلد ثابت کرد که شرط لازم برای پایداری این شبکه این است که  $w$  یک ماتریس متقارن باشد. علاوه بر این هاپفیلد نشان داد که نقطه تعادل شبکه، نقطه مینیمم محلی تابع لیاپانوف زیر است.

$$E(x) = -\frac{1}{2} x^T w x - x^T \theta + \sum_{j=1}^n \alpha_j \int_0^1 f^{-1}(z) dz. \quad (5.1)$$

بردار  $x = [x_1, \dots, x_n]$  بردار خروجی شبکه است که هر درایه آن  $x_i = f(u_i)$ . همانطور که قبلاً ذکر شد  $f(u_j) = \tanh(\beta u_j) = \frac{e^{\beta u_j} - e^{-\beta u_j}}{e^{\beta u_j} + e^{-\beta u_j}}$ ، هرگاه  $\beta \rightarrow \pm\infty$  در این صورت  $f(u_j) \rightarrow \pm 1$  و لذا  $E(x)$  تابعی کراندار خواهد بود.

به شبکه هاپفیلد به مشابه سیستم گرادیان نیز می‌توان نگاه کرد.

$$\frac{du}{dt} = -\rho \nabla E(x), \quad (6.1)$$

که در آن  $\rho = \text{diag}(\frac{1}{C_1}, \dots, \frac{1}{C_n})$ . نقطه ساکن سیستم (۶.۱) مینیمم محلی برای  $E(x)$  است زیرا از  $\frac{du}{dt} = 0$  نتیجه می‌شود که  $\nabla E(x) = 0$  است. حال ثابت می‌کنیم که نقطه ساکن سیستم (۶.۱) به طور مجانبی پایدار است. برای این منظور باید در همسایگی نقطه ساکن سیستم، یک تابع لیاپانوف به دست آوریم.

قضیه ۵.۱.۱. نقطه ساکن سیستم  $\dot{x}(t) = f(x(t))$  پایدار است اگر تابع لیاپانوف متناظر با این سیستم وجود داشته باشد.

□

برهان. به مرجع [۱۱] مراجعه کنید.

قضیه ۶.۱.۱. نقطه ساکن سیستم دینامیکی (۶.۱) به طور مجانبی پایدار است.

برهان. [۵۰] برای هر نقطه ساکن  $x^*$  همسایگی مانند  $\sum$  وجود دارد که تابع  $V(x) = E(x) - E(x^*)$  روی این همسایگی به طور پیوسته مشتق پذیر است بدلیل اینکه  $E(x)$  به طور پیوسته مشتق پذیر است. همچنین  $V(x)$  اکیدا مثبت است و  $V(x^*) = 0$ .

همچنین

$$\frac{dV}{dt} = \frac{dE}{dt} = \nabla E(x) \dot{x}(t) = \sum_{j=1}^n \frac{\partial E}{\partial x_j} \cdot \frac{\partial x_j}{\partial t} = -C \sum_{j=1}^n \frac{du_j}{dt} \cdot \frac{dx_j}{dt} = -C \sum_{j=1}^n \frac{dx_j}{du_j} \cdot \left(\frac{du_j}{dt}\right)^2 \leq 0.$$

همواره مثبت است چون  $x_i = f(u_i)$  است و همانطور که قبلا ذکر شد  $f$  یک تابع غیر نزولی است. لذا نتیجه می‌گیریم که  $V(x)$  یک تابع لیاپانوف است، بنا به قضیه (۵.۱.۱) نقطه ساکن سیستم (۶.۱) به طور مجانبی پایدار است.  $\square$

به طور کلی اگر بخواهیم شبکه هاپفیلد را از دید نرم افزاری بیان کنیم، می‌توان اینطور گفت که معادله دینامیکی (۶.۱) وضعیت شبکه نسبت به زمان را مشخص می‌کند. متناظر با این سیستم دینامیکی تابع انرژی  $E(x)$  به گونه‌ای تعریف می‌شود که یک تابع لیاپانوف باشد لذا این اطمینان حاصل می‌شود که شبکه همواره همگرا باشد. همچنین نقاط ساکن  $\frac{du}{dt} = -\rho \nabla E(x)$  نقاط مینیمم برای تابع نزولی  $E(x)$  می‌باشد که همان الگوهای موردنظر مساله است. با وارد کردن یک الگوی ناشناس اولیه، شبکه آنقدر تکرار می‌شود تا در همسایگی یکی از این الگوهای مشخص قرار بگیرد به همین دلیل گفته می‌شود که شبکه هاپفیلد قابلیت ذخیره سازی الگو را دارد.



## فصل ۲

### مقدمه ای بر کنترل بهینه

#### ۱.۲ مقدمه

روش‌های مختلفی که برای طراحی دستگاه‌های کنترل بکار برده می‌شوند، معمولاً روش‌های سعی و خطا هستند که در آن‌ها برای تعیین پارامترهای طراحی دستگاه، روش‌های مختلف تحلیلی، به صورت متوالی مورد استفاده قرار می‌گیرند. نحوه عملکرد دستگاه معمولاً بر حسب مشخصه‌های فرکانسی نظیر حد فاز، حد دامنه و پهنای باند بیان می‌شوند. لیکن با این روش، در مورد دستگاه‌هایی با چند ورودی و چند خروجی که نیازهای تکنولوژیکی امروزه را برآورد می‌نمایند، باید معیارها یا نحوه عملکردهای گوناگونی صادق باشند. بعنوان مثال، طرح دستگاه کنترل موقعیت هواپیما که مصرف سوخت را نیز حداقل کند با استفاده از روش‌های متعارف امکان پذیر نیست. روش جدید و مستقیم طرح چنین دستگاه‌های پیچیده‌ای که کنترل بهینه<sup>۱</sup> نامیده می‌شود، با توسعه کامپوترهای دیجیتال، امکان پذیر شده است. هدف دستگاه کنترل بهینه تعیین متغیرهای کنترلی است که در محدودیت‌ها یا قیود مساله صدق کرده و نحوه عملکرد یا معیار معینی را حداقل یا حداکثر نمایند.

#### ۲.۲ تنظیم صورت مساله

در مدلسازی مسائل کنترل بهینه به موارد زیر نیاز داریم:

۱- بیان ریاضی یا مدل دستگاهی که باید کنترل شود،

۲- بیان محدودیت‌های فیزیکی،

۳- تعیین نحوه عملکرد دستگاه.

---

<sup>۱</sup>optimal control

## ۱.۲.۲ مدل ریاضی

جزء قابل توجهی از هر مساله کنترلی، مدلسازی دستگاه است. هدف، بدست آوردن ساده ترین بیان ریاضی است که پاسخ دستگاه فیزیکی را به تمام ورودی‌های مورد نظر بطور مناسب پیش بینی کند. دستگاه با معادلات دیفرانسیل معمولی را در نظر بگیرید. اگر  $x_1(t), \dots, x_n(t)$  متغیرهای وضعیت دستگاه در زمان  $t$  و  $u_1(t), \dots, u_m(t)$  متغیرهای کنترلی در زمان  $t$  باشند، دستگاه با معادلات دیفرانسیل مرتبه‌ی اول به شکل زیر بیان می‌گردد:

$$\begin{cases} \dot{x}_1(t) = g_1(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t), t), \\ \vdots \\ \dot{x}_n(t) = g_n(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t), t), \end{cases}$$

بردار  $x(t) = (x_1(t), \dots, x_n(t))$  بردار وضعیت دستگاه و بردار  $u(t) = (u_1(t), \dots, u_m(t))$  بردار کنترل نامیده می‌شود. لذا داریم:

$$\dot{x}(t) = g(x(t), u(t), t). \quad (1.2)$$

## ۲.۲.۲ محدودیت‌ها یا قيود فیزیکی

پس از انتخاب مدل ریاضی دستگاه، باید محدودیت‌های فیزیکی را برای وضعیت‌ها و کنترل‌ها تعریف نمود.

**تعریف ۱.۲.۲.** کنترلی که در تمام مدت  $[t_0, t_f]$  در محدودیت‌های کنترل صدق نماید، کنترل قابل قبول<sup>۲</sup> نامیده شده و مجموعه‌ی این کنترل‌های قابل قبول با  $U$  نشان داده می‌شود.

**تعریف ۲.۲.۲.** منحنی مسیر متغیر وضعیت که در تمام مدت  $[t_0, t_f]$  در محدودیت‌های متغیر وضعیت صدق می‌کند، منحنی مسیر قابل قبول<sup>۳</sup> نامیده شده و مجموعه‌ی این منحنی‌های قابل قبول با  $X$  نشان داده می‌شود.

## ۳.۲.۲ تابعی معیار

برای ارزیابی کمی عملکرد دستگاه، باید تابعی معیار انتخاب گردد. دستگاه بهینه، دستگاهی است که این تابعی معیار<sup>۴</sup> را حداکثر یا حداقل نماید. عملکرد دستگاه با تابعی معیار به شکل زیر ارزیابی می‌شود:

$$J = h(x(t_f), t_f) + \int_{t_0}^{t_f} f_0(x(t), u(t), t) dt, \quad (2.2)$$

که در آن  $t_0$  و  $t_f$  به ترتیب زمان شروع و زمان نهایی بوده و  $f_0$  و  $h$  توابع اسکالر هستند.

<sup>۲</sup>Admissible Control

<sup>۴</sup>Functional criteria

<sup>۳</sup>Admissible Trajectory



## ۳.۲ مساله‌ی کنترل بهینه

شکل کلی یک مساله‌ی کنترل بهینه به صورت زیر است:

$$\begin{aligned} & \text{minimize } J = h(x(t_f), t_f) + \int_{t_0}^{t_f} f_0(x(t), u(t), t) dt, \\ & \text{subject to} \\ & \dot{x} = g(x(t), u(t), t), \\ & x(t_0) = x_0. \end{aligned} \quad (3.2)$$

که  $u^*$  کنترل بهینه و  $x^*$  منحنی مسیر بهینه برای مساله (۳.۲) می‌باشد. دستگاه‌ها را براساس معادلات وضعیت آنها به شکل زیر دسته بندی می‌کنیم:

- دستگاه‌های خطی متغیر با زمان<sup>۵</sup>
- دستگاه‌های غیر خطی متغیر با زمان<sup>۶</sup>
- دستگاه‌های خطی ثابت با زمان<sup>۷</sup>
- دستگاه‌های غیر خطی ثابت با زمان<sup>۸</sup>

تعریف ۱.۳.۲. سیستم دینامیکی زیر را در نظر بگیرید:

$$\dot{x} = f(x(t)), \quad x(t_0) = x_0 \in \mathbb{R}^n,$$

که در آن  $f$  یک تابع از  $\mathbb{R}^n$  به  $\mathbb{R}^n$  می‌باشد.  $x^*$  یک نقطه تعادل<sup>۹</sup> سیستم دینامیکی فوق نامیده می‌شود اگر  $f(x^*) = 0$ .

تعریف ۲.۳.۲. دستگاه (۱.۲) را برای  $t \geq t_0$  با وضعیت اولیه  $x(t_0) = x_0$  در نظر می‌گیریم. اگر زمان محدود  $t_1 \geq t_0$  و کنترل  $u(t)$ ,  $t \in [t_0, t_1]$  وجود داشته باشد که در زمان  $t_1$  وضعیت  $x_0$  را به نقطه تعادل منتقل کند، آنگاه وضعیت  $x_0$  را در زمان  $t_0$  کنترل پذیر<sup>۱۰</sup> می‌نامیم.

تعریف ۳.۳.۲. اگر بتوانیم با مشاهده‌ی خروجی  $y(t)$  در بازه‌ی زمانی محدود  $[t_0, t_f]$  وضعیت  $x(t_0) = x_0$  را تعیین کنیم، وضعیت  $x_0$  در زمان  $t_0$  مشاهده پذیر<sup>۱۱</sup> نامیده می‌شود.

<sup>۵</sup>Linear time varying

<sup>۶</sup>Nonlinear time varying

<sup>۷</sup>Linear time invariant

<sup>۸</sup>Nonlinear time invariant

<sup>۹</sup>Equilibrium Point

<sup>۱۰</sup>Controllable

<sup>۱۱</sup>Observables

## ۴.۲ حساب تغییرات

حساب تغییرات<sup>۱۲</sup> رشته‌ای از ریاضیات است که در حل مسائل بهینه سازی بسیار مفید می‌باشد. دیدو<sup>۱۳</sup>، ملکه کارتاژ اولین کسی بود که با چنین مساله‌ای مواجه شد. به او قول داده شده بود که تمام زمینی را که با قطعه طنابی محدود کند، می‌تواند تصاحب نماید. برای این منظور او طنابی را به رشته‌های نازک تقسیم کرد و آنها را به هم گره زد، آنگاه مساله این بود که بین سطوح هندسی مختلف با محیط ثابت کدام یک بزرگترین مساحت را دارد.

تاریخچه مساله حساب تغییرات مربوط به یونان قدیم است، ولی پیشرفت زیاد اروپا در این زمینه تا قرن هفده میلادی به طول انجامید. مساله تاریخی جالب دیگر، مساله بارچيستوچرون<sup>۱۴</sup> است و در سال ۱۶۹۶ توسط برنولی ارائه گردید. او به یک منحنی بین دو نقطه ثابت  $A$  و  $B$  در صفحه قائم احتیاج داشت که جسمی در آن (تحت نیروی ثقل و بدون اصطکاک) بلغزد و در حداقل زمان به پائین بیاید. جواب این مساله بدیهی نبود. خط مستقیم بین  $A$  و  $B$  جواب مساله نیست، این مطلب را بطور تجربی با سردادن مهره‌های سیمی به شکل‌های مناسب می‌توان دریافت. در هر دو مساله دیدو و بارچيستوچرون، منحنی‌هایی که مورد جستجو قرار می‌گیرند باعث می‌شوند که معیاری مقدارنهایت خود را بپذیرد. در نتیجه برخی از مسائل مورد توجه مربوط به شاخه حساب تغییرات عبارت‌اند از:

- پیدا کردن فاصله بین دو نقطه،
  - پیدا کردن یک منحنی بین همه منحنی‌های واقع در صفحه  $xy$  بطوریکه سطح حاصل از دوران آن حول محور  $x$  حداقل شود.
  - پیدا کردن خمی که دو نقطه را وصل کند بطوریکه اگر گلوله‌ای از نقطه بالاتر به سمت پائین حرکت کند زمان طی شده حداقل شود.
  - درحقیقت حساب تغییرات عبارتست از تعیین یک تابع خاص که یک تابعی معین به نام تابعی معیار را کمینه کند.
- تعریف ۱.۴.۲. تابعی  $J$  قانون تناظری است که به هر تابع  $x$  در دسته معین  $\Omega$  یک عدد حقیقی منحصر به فرد را نسبت می‌دهد.

### ۱.۴.۲ تقریب توابع

تعریف ۲.۴.۲. اگر  $q$  و  $q + \Delta q$  عناصری باشند که تابع  $f$  بر آنها تعریف شده باشد، آنگاه نمو<sup>۱۵</sup>  $f$  با  $\Delta f$  نشان داده می‌شود و عبارتست از:

$$\Delta f \triangleq f(q + \Delta q) - f(q).$$

<sup>۱۲</sup>Calculus of Variations

<sup>۱۴</sup>Barchistochrone

<sup>۱۳</sup>Dido

<sup>۱۵</sup>Increment

تعریف ۳.۴.۲. اگر  $x$  و  $x + \Delta x$  عناصری باشند که تابع  $J$  بر آنها تعریف شده باشد، آنگاه نمودار  $J$  با  $\Delta J$  نشان داده می‌شود و عبارتست از:

$$\Delta J \triangleq J(x + \Delta x) - J(x).$$

تعریف ۴.۴.۲. نمودار تابعی را می‌توان بصورت زیر تعریف کرد:

$$\Delta J(x, \delta x) = \delta J(x, \delta x) + g(x, \delta x) \|\delta x\|,$$

که  $\delta J$  قسمت خطی بر حسب  $\delta x$  می‌باشد. اگر  $\lim_{\|\delta x\| \rightarrow 0} g(x, \delta x) = 0$ ، آنگاه  $J$  را دیفرانسیل پذیر<sup>۱۶</sup> گویند و  $\delta J$  تغییرات تابعی  $J$  نامیده می‌شود.

قضیه ۵.۴.۲. (قضیه اساسی حساب تغییرات). اگر  $x^*$  یک منحنی اکسترمم باشد، تغییرات  $J$  روی  $x^*$  باید صفر شود، یعنی به ازای هر  $\delta x$  قابل قبول  $\delta J(x^*, \delta x) = 0$ .

برهان. به مرجع [۲۵] مراجعه شود. □

لم ۶.۴.۲. (لم اساسی حساب تغییرات). اگر تابع  $h$  پیوسته بوده و برای هر تابع پیوسته مانند  $\delta x$  در محدوده  $[t_0, t_f]$  داشته باشیم:

$$\int_{t_0}^{t_f} h(t) \delta x(t) dt = 0,$$

آنگاه  $h$  باید در تمام محدوده  $[t_0, t_f]$  صفر باشد.

برهان. به مرجع [۲۵] مراجعه شود. □

## ۲.۴.۲ ساده ترین مساله تغییراتی

اگر  $x$  یک تابع اسکالر با مشتقات مرتبه اول پیوسته باشد، می‌خواهیم تابع  $x^*$  را چنان پیدا کنیم که تابعی:

$$J = \int_{t_0}^{t_f} f_0(x(t), \dot{x}(t), t) dt,$$

در  $x^*$  دارای اکسترمم نسبی<sup>۱۷</sup> باشد.

فرض می‌کنیم  $f_0$  دارای مشتقات نسبی مرتبه اول و دوم پیوسته نسبت به هریک از متغیرهای خود باشد.  $t_0$  و  $t_f$  ثابت و نقاط انتهایی منحنی به صورت  $x_0$  و  $x_f$  مشخص هستند. در جستجوی یک منحنی هستیم (در صورت وجود) که  $J(x)$  را به اکسترمم خود ببرد.

تغییرات  $\delta J(x, \delta x)$  به صورت زیر به دست می‌آید:

$$\delta J(x, \delta x) = \int_{t_0}^{t_f} \left\{ \left[ \frac{\partial f_0}{\partial x}(x(t), \dot{x}(t), t) \right] \delta x(t) + \left[ \frac{\partial f_0}{\partial \dot{x}}(x(t), \dot{x}(t), t) \right] \delta \dot{x} \right\} dt,$$

اگر منحنی اکسترممی مانند  $x^*$  را در نظر بگیریم و با انتگرال‌گیری جز به جز از قسمت دوم انتگرال و اعمال قضیه اساسی حساب تغییرات داریم:

$$\delta J(x, \delta x) = 0 = \int_{t_0}^{t_f} \left\{ \frac{\partial f_0}{\partial x}(x^*(t), \dot{x}^*(t), t) - \frac{d}{dt} \left[ \frac{\partial f_0}{\partial \dot{x}}(x^*(t), \dot{x}^*(t), t) \right] \right\} \delta x(t) dt,$$

<sup>۱۶</sup>Differentiable

<sup>۱۷</sup>Relative Extremum

حال با اعمال لم اساسی حساب تغییرات شرط لازم برای اینکه  $x^*$  به ازای هر  $t \in [t_0, t_f]$  اکسترمم باشد به صورت زیر به دست می‌آید:

$$\frac{\partial f_0}{\partial x}(x^*(t), \dot{x}^*(t), t) - \frac{d}{dt} \left[ \frac{\partial f_0}{\partial \dot{x}}(x^*(t), \dot{x}^*(t), t) \right] = 0,$$

این معادله، معادله‌ی اوایلر<sup>۱۸</sup> نامیده می‌شود.

چون معادله‌ی اوایلر شرط لازم است، لذا برای تشخیص اینکه  $x^*$  منحنی بهینه است، باید بررسی بیشتری انجام داد. در صورتی که  $t_f$  مشخص و  $x_f$  آزاد باشد، شرایط لازم برای تابع  $x^*$  که اکسترمم تابعی  $J(x) = \int_{t_0}^{t_f} f_0(x(t), \dot{x}(t), t) dt$  باشد، عبارتند از برقراری معادله‌ی اوایلر و همچنین برقراری رابطه‌ی  $\frac{\partial f_0}{\partial \dot{x}}(x^*(t_f), \dot{x}^*(t_f), t_f) = 0$ . این معادله شرط مرزی طبیعی<sup>۱۹</sup> نامیده می‌شود.

اگر در مساله‌ای  $t_f$  آزاد و  $x_f$  مشخص باشد شرایط لازم عبارتند از برقراری معادله‌ی اوایلر و برقراری معادله‌ی  $f_0(x^*(t_f), \dot{x}^*(t_f), t_f) - \left[ \frac{\partial f_0}{\partial \dot{x}}(x^*(t_f), \dot{x}^*(t_f), t_f) \right] \dot{x}^*(t_f) = 0$  در مسائلی که  $x_f$  و  $t_f$  هر دو آزاد هستند، دو حالت زیر را در نظر می‌گیریم:

(۱)  $x_f$  و  $t_f$  مرتبط نمی‌باشند. در این صورت شرایط لازم عبارتند از برقراری معادله‌ی اوایلر و برقراری معادله‌ی  $f_0(x^*(t_f), \dot{x}^*(t_f), t_f) = 0$

(۲)  $x_f$  و  $t_f$  مرتبط می‌باشند. به عنوان مثال  $x_f = \theta(t_f)$ . در این حالت نیز باید علاوه بر برقراری معادله‌ی اوایلر، معادله‌ی

$$\left[ \frac{\partial f_0}{\partial \dot{x}}(x^*(t_f), \dot{x}^*(t_f), t_f) \right] \left[ \frac{d\theta}{dt}(t_f) - \dot{x}^*(t_f) \right] + f_0(x^*(t_f), \dot{x}^*(t_f), t_f) = 0$$

که شرط اربیبی<sup>۲۰</sup> نامیده می‌شود، برقرار باشد.

### ۳.۴.۲ شرایط لازم برای کنترل بهینه

هدف تعیین کنترل قابل قبول مانند  $u^*$  است که باعث شود دستگاه (۱.۲) با طی کردن منحنی مسیر قابل قبول  $x^*$ ، تابعی معیار زیر را کمینه کند.

$$J(u) = h(x(t_f), t_f) + \int_{t_0}^{t_f} f_0(x(t), u(t), t) dt,$$

با فرض دیفرانسیل پذیری تابع  $h$  و استفاده از قاعده‌ی زنجیره‌ای دیفرانسیل گیری،  $J(u)$  به صورت زیر به دست می‌آید:

$$J(u) = \int_{t_0}^{t_f} \left\{ f_0(x(t), u(t), t) + \left[ \frac{\partial h}{\partial x}(x(t), t) \right]^T \dot{x}(t) + \frac{\partial h}{\partial t}(x(t), t) \right\} dt. \quad (۴.۲)$$

<sup>۱۸</sup>Euler

<sup>۲۰</sup>Transversality condition

<sup>۱۹</sup>Natural boundary condition

با معرفی ضرایب لاگرانژ  $p_n(t), \dots, p_1(t)$  و تعریف تابع  $g_a(x(t), \dot{x}(t), u(t), p(t), t)$  به صورت

$$g_a(x(t), \dot{x}(t), u(t), p(t), t) \triangleq f_0(x(t), u(t), t) + p^T(t) [g(x(t), u(t), t) - \dot{x}(t)] \\ + \left[ \frac{\partial h}{\partial x}(x(t), t) \right]^T \dot{x}(t) + \frac{\partial h}{\partial t}(x(t), t),$$

داریم

$$J_a(u) = \int_{t_0}^{t_f} g_a(x(t), \dot{x}(t), u(t), p(t), t) dt.$$

حال با اعمال قضیه اساسی حساب تغییرات و تعریف تابع  $\mathcal{H}(x(t), u(t), p(t), t)$  به نام تابع هامیلتونین به صورت زیر، شرایط لازم به دست می‌آیند.

$$\mathcal{H}(x(t), u(t), p(t), t) \triangleq f_0(x(t), u(t), t) + p^T \cdot g(x(t), u(t), t). \quad (۵.۲)$$

شرایط لازم عبارتند از:

$$\frac{\partial \mathcal{H}}{\partial x}(x^*(t), u^*(t), p^*(t), t) = -\dot{p}(t), \quad (۶.۲)$$

$$\frac{\partial \mathcal{H}}{\partial u}(x^*(t), u^*(t), p^*(t), t) = 0, \quad (۷.۲)$$

$$\frac{\partial \mathcal{H}}{\partial p}(x^*(t), u^*(t), p^*(t), t) = \dot{x}(t), \quad (۸.۲)$$

$$\left[ \frac{\partial h}{\partial x}(x^*(t_f), t_f) - p^*(t_f) \right]^T \delta x_f + \left[ \mathcal{H}(x^*(t_f), u^*(t_f), p^*(t_f), t_f) + \frac{\partial h}{\partial t}(x^*(t_f), t_f) \right] \delta t_f = 0. \quad (۹.۲)$$

## ۵.۲ اصل حداقل یابی پونتریگین

در دستگاه‌های واقعی اغلب محدودیت‌هایی روی کنترل‌ها و وضعیت‌های قابل قبول وجود دارد. می‌خواهیم ببینیم این محدودیت‌ها چگونه بر شرایط لازم که با فرض غیر مقید بودن مقادیر کنترل‌های قابل قبول به دست آمده‌اند، تاثیر می‌گذارد.

طبق تعریف، اگر برای کنترل‌های قابل قبولی که به اندازه‌ی کافی نزدیک  $u^*$  هستند، داشته باشیم

$$J(u) - J(u^*) = \Delta J \geq 0$$

آنگاه کنترل  $u^*$  باعث می‌شود تابعی  $J$  کمینه نسبی داشته باشد.

اگر  $u = u^* + \delta u$ ، نمو  $J$  با رابطه‌ی زیر بیان می‌شود:

$$\Delta J(u^*, \delta u) = \delta J(u^*, \delta u) + \text{جملات مراتب بالاتر}$$

$\delta J$  بر حسب  $\delta u$  خطی است و جملات مراتب بالاتر زمانی که  $\delta u$  به سمت صفر میل می‌کند، همگی به سمت صفر میل می‌نمایند.

اگر کنترل نامقید باشد، با استفاده از این که  $\delta J$  بر حسب  $\delta u$  خطی است و  $\delta u$  می‌تواند به صورت اختیاری تغییر کند می‌توان نشان داد شرایط لازم برای اینکه  $u^*$  منحنی کنترل اکسترمم باشد این است

که تغییرات  $\delta J(u^*, \delta u)$  برای تمام  $\delta u$  های قابل قبول که نورم کوچکی دارند صفر شود. اما چون فرض شده است که کنترل های قابل قبول محدود باشند،  $\delta u$ ، فقط زمانی اختیاری است که کنترل اکسترمم در تمام بازه‌ی زمانی  $[t_0, t_f]$  بین دو حد تعیین شده باشد.

با بررسی تمام تغییرات قابل قبول با  $\|\delta u\|$  به اندازه‌ی کافی کوچک، بطوریکه علامت  $\Delta J$  توسط  $\delta u$  تعیین می‌شود، نتیجه می‌شود شرط لازم برای اینکه  $u^*$  تابعی  $J$  را حداقل کند، این است که:

$$\delta J(u^*, \delta u) \geq 0$$

حال باید ببینیم این تغییرات چگونه بر شرایط لازمی که با فرض غیر مقید بودن مقادیر کنترل های قابل قبول به دست آمده‌اند، تاثیر می‌گذارد. نمو  $J$  به صورت زیر است:

$$\begin{aligned} \Delta J(u^*, \delta u) = & \left[ \frac{\partial h}{\partial x}(x^*(t_f), t_f) - p^*(t_f) \right]^T \delta x_f \\ & + \left[ \mathcal{H}(x^*(t_f), u^*(t_f), p^*(t_f), t_f) + \frac{\partial h}{\partial t}(x^*(t_f), t_f) \right] \delta t_f \\ & + \int_{t_0}^{t_f} \left\{ \left[ p^*(t) + \frac{\partial \mathcal{H}}{\partial x}(x^*(t), u^*(t), p^*(t), t) \right]^T \delta x(t) \right. \\ & + \left[ \frac{\partial \mathcal{H}}{\partial u}(x^*(t), u^*(t), p^*(t), t) \right]^T \delta u(t) \\ & \left. + \left[ \frac{\partial \mathcal{H}}{\partial p}(x^*(t), u^*(t), p^*(t), t) - \dot{x}^*(t) \right]^T \delta p(t) \right\} dt + \text{جملات مراتب بالاتر} \end{aligned}$$

اگر معادلات حالت برقرار باشند و  $p^*(t)$  چنان انتخاب شود که ضرایب  $\delta x(t)$  در انتگرال صفر شوند و معادلات شرایط حدی برقرار باشند داریم:

$$\Delta J(u^*, \delta u) = \int_{t_0}^{t_f} \left[ \frac{\partial \mathcal{H}}{\partial u}(x^*(t), u^*(t), p^*(t), t) \right]^T \delta u(t) dt + \text{جملات مراتب بالاتر}$$

از طرفی می‌دانیم که

$$\begin{aligned} \left[ \frac{\partial \mathcal{H}}{\partial u}(x^*(t), u^*(t), p^*(t), t) \right]^T \delta u(t) = & \mathcal{H}(x^*(t), u^*(t), \delta u(t), p^*(t), t) \\ & - \mathcal{H}(x^*(t), u^*(t), p^*(t), t) \end{aligned}$$

در نتیجه داریم:

$$\begin{aligned} \Delta J(u^*, \delta u) = & \int_{t_0}^{t_f} \left\{ \mathcal{H}(x^*(t), u^*(t), \delta u(t), p^*(t), t) \right. \\ & \left. - \mathcal{H}(x^*(t), u^*(t), p^*(t), t) \right\} dt + \text{جملات مراتب بالاتر} \end{aligned}$$

بنابراین برای اینکه  $u^*$  کنترل حداقل کننده‌ای باشد، باید به ازای هر مقدار قابل قبول  $\delta u$  داشته باشیم

$$\int_{t_0}^{t_f} \left\{ \mathcal{H}(x^*(t), u^*(t), \delta u(t), p^*(t), t) - \mathcal{H}(x^*(t), u^*(t), p^*(t), t) \right\} dt \geq 0$$

که ایجاب می‌کند برای هر  $\delta u$  قابل قبول داشته باشیم

$$\mathcal{H}(x^*(t), u^*(t), \delta u(t), p^*(t), t) \geq \mathcal{H}(x^*(t), u^*(t), p^*(t), t)$$

بنابراین نتیجه می‌گیریم که شرط لازم برای اینکه  $u^*$  تابعی  $J$  را حداقل کند این است که به ازای تمام کنترل‌های قابل قبول داشته باشیم:

$$\mathcal{H}(x^*(t), u^*(t), p^*(t), t) \leq \mathcal{H}(x^*(t), u(t), p^*(t), t)$$

این معادله نمایانگر این مطلب است که کنترل بهینه باید هامیلتونین را حداقل کند و به اصل حداقل‌یابی پونتری‌اگین<sup>۲۱</sup> معروف می‌باشد.

## ۶.۲ سیستم‌های تنظیم کننده‌ی خطی

مسائل کنترل بهینه درجه‌ی دوم خطی<sup>۲۲</sup> نقش مهمی در حساب تغییرات ایفا می‌کند. یکی از انواع مسائل کنترل بهینه درجه‌ی دوم سیستم‌های تنظیم کننده خطی<sup>۲۳</sup> است. نشان خواهیم داد که برای مسائل تنظیم کننده خطی، قانون کنترل بهینه می‌تواند بصورت یک تابع خطی، متغیر با زمان از متغیرهای حالت بدست آید. تحت شرایط خاصی که بحث خواهیم کرد، قانون کنترل بهینه غیر متغیر با زمان خواهد شد. در پایان نیز یک مثال از این نوع سیستم به روش تشریحی حل می‌کنیم. سیستمی با معادلات خطی زیر که ممکن است دارای ضرایب متغیر با زمان باشد، بصورت زیر بیان شده است:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t). \quad (10.2)$$

تابع هدفی که باید حداقل شود بصورت زیر است:

$$J = \frac{1}{2} x^T(t_f) H x(t_f) + \frac{1}{2} \int_{t_0}^{t_f} [x^T(t) Q(t) x(t) + u^T(t) R(t) u(t)] dt, \quad (11.2)$$

که در آن زمان نهایی  $t_f$  ثابت<sup>۲۴</sup> است،  $H$  و  $Q$  ماتریس‌های حقیقی، متقارن نیمه معین مثبت و  $R$  ماتریس حقیقی، متقارن معین مثبت می‌باشد. فرض می‌شود که حالت‌ها و کنترل‌ها محدود نیستند و  $x(t_f)$  آزاد<sup>۲۵</sup> یا غیرمعین است. می‌خواهیم بردار حالت را بدون صرف نیروی کنترل زیادی نزدیک مبدا قرار دهیم.

برای بدست آوردن جواب دقیق مساله (۱۱.۲) - (۱۰.۲) هامیلتونین<sup>۲۶</sup> را بصورت زیر تشکیل می‌دهیم:

$$\mathcal{H}(x(t), u(t), p(t), t) = \frac{1}{2} x^T(t) Q(t) x(t) + \frac{1}{2} u^T(t) R(t) u(t) + p^T(t) (A(t)x(t) + B(t)u(t)).$$

<sup>۲۱</sup>Pontryagin's Minimum Principle

<sup>۲۲</sup>Linear quadratic optimal control problems

<sup>۲۳</sup>Linear regulator systems

<sup>۲۴</sup>Constant

<sup>۲۵</sup>Free

<sup>۲۶</sup>Hamiltonian

شرایط لازم برای بهینگی عبارتند از:

$$\frac{\partial \mathcal{H}}{\partial p} = \dot{x} \Rightarrow \dot{x}(t) = A(t)x(t) + B(t)u(t), \quad (12.2)$$

$$\frac{\partial \mathcal{H}}{\partial x} = -\dot{p}(t) \Rightarrow Q(t)x(t) + A^T(t)p(t) = -\dot{p},$$

$$\frac{\partial \mathcal{H}}{\partial u} = 0 \Rightarrow R(t)u(t) + B^T(t)p(t) = 0. \quad (13.2)$$

معادله (۱۳.۲) می‌تواند برای  $u$  حل شود، لذا

$$u(t) = -R^{-1}(t)B^T(t)p(t). \quad (14.2)$$

چون  $R$  ماتریس مثبت معین است  $R^{-1}$  وجود خواهد داشت. جایگزینی معادله (۱۴.۲) در معادله (۱۲.۲) ایجاب می‌کند که

$$\dot{x}(t) = A(t)x(t) - B(t)R^{-1}(t)B^T(t)p(t).$$

باشد بنابراین مجموعه‌ای از  $2n$  معادله دیفرانسیل خطی همگن خواهیم داشت:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{p}(t) \end{bmatrix} = \begin{bmatrix} A(t) & -B(t)R^{-1}(t)B^T(t) \\ -Q(t) & -A^T(t) \end{bmatrix} \begin{bmatrix} x(t) \\ p(t) \end{bmatrix}. \quad (15.2)$$

پاسخ این معادلات به شکل زیر است:

$$\begin{bmatrix} x(t_f) \\ p(t_f) \end{bmatrix} = \phi(t_f, t) \begin{bmatrix} x(t) \\ p(t) \end{bmatrix},$$

که در آن  $\phi$  ماتریس انتقال سیستم<sup>۲۷</sup> (۱۵.۲) می‌باشد. با مجزا کردن ماتریس انتقال داریم:

$$\begin{bmatrix} x(t_f) \\ p(t_f) \end{bmatrix} = \begin{bmatrix} \phi_{11}(t_f, t) & \phi_{12}(t_f, t) \\ \phi_{21}(t_f, t) & \phi_{22}(t_f, t) \end{bmatrix} \begin{bmatrix} x(t) \\ p(t) \end{bmatrix}, \quad (16.2)$$

که  $\{\phi_{11}, \phi_{12}, \phi_{21}, \phi_{22}\}$  ماتریس‌های  $n \times n$  می‌باشند. از معادلات شرایط حدی [۲۵]، درمی‌یابیم که

$$p(t_f) = Hx(t_f).$$

با جایگزینی این مقدار  $p(t_f)$  در معادله (۱۶.۲) خواهیم داشت:

$$x(t_f) = \phi_{11}(t_f, t)x(t) + \phi_{12}(t_f, t)p(t),$$

$$Hx(t_f) = \phi_{21}(t_f, t)x(t) + \phi_{22}(t_f, t)p(t).$$

با جایگزینی معادله بالایی در معادله پایینی داریم:

$$H\phi_{11}(t_f, t)x(t) + H\phi_{12}(t_f, t)p(t) = \phi_{22}(t_f, t)p(t) + \phi_{21}(t_f, t)x(t).$$

<sup>۲۷</sup>Transition matrix



که چون برای  $p(t)$  حل شود، داریم:

$$p(t) = [\phi_{22}(t_f, t) - H\phi_{12}(t_f, t)]^{-1} [H\phi_{11}(t_f, t) - \phi_{21}(t_f, t)]x(t). \quad (17.2)$$

کالمن<sup>۲۸</sup> [۳۸] نشان داده است که به ازای هر  $t \in [t_0, t_f]$  ماتریس معکوس مورد لزوم معادله فوق وجود دارد. معادله (۱۷.۲) بصورت زیر نیز می‌تواند نوشته شود:

$$p(t) = K(t)x(t). \quad (18.2)$$

که بدین معنی است که  $p(t)$  تابع خطی از حالت‌های سیستم است.  $K$  یک ماتریس  $n \times n$  است. در حقیقت  $K$  به  $t_f$  نیز بستگی دارد، لیکن  $t_f$  مشخص است. با جایگزینی در معادله (۱۴.۲) داریم:

$$\begin{aligned} u(t) &= -R^{-1}(t)B^T(t)K(t)x(t), \\ &= F(t)x(t). \end{aligned} \quad (19.2)$$

معادله فوق نشان می‌دهد که قانون کنترل بهینه، قانونی است خطی، متغیر با زمان و ترکیبی از حالت‌های سیستم است. متذکر می‌شویم که حتی اگر دستگاه ثابت باشد ماتریس بازخورد  $F$  متغیر با زمان است. علاوه، اندازه‌ی همه‌ی متغیرهای حالت باید در دسترس باشند تا قانون کنترل بهینه انجام شود. شکل (۱۰.۲) دستگاه کنترل کننده بهینه را نشان می‌دهد.

برای تعیین ماتریس ضریب تقویت بازخورد<sup>۲۹</sup>  $F$ ، احتیاج به ماتریس انتقال سیستم (۱۵.۲) می‌باشد. اگر تمام ماتریس‌های بکارگرفته‌شده  $\{A, B, R, Q\}$  ثابت باشند، ماتریس انتقال مورد نیاز با برآورد تبدیل معکوس لاپلاس ماتریس

$$\left\{ sI - \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \right\}^{-1},$$

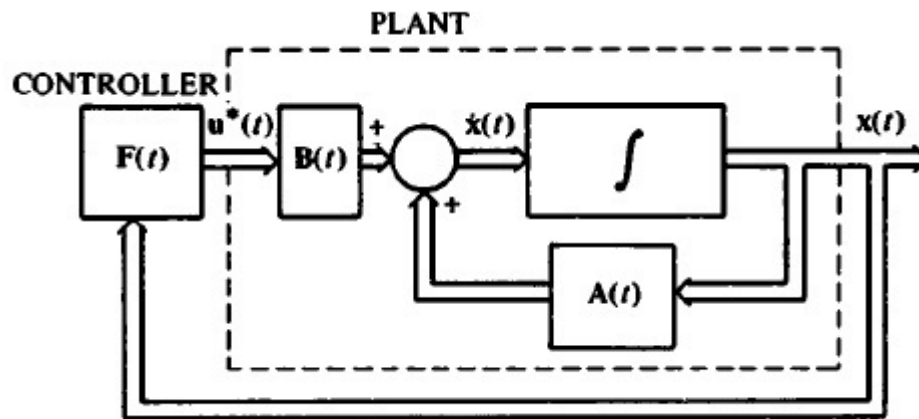
و جایگزینی  $(t_f - t)$  برای  $t$  بدست می‌آید. متأسفانه وقتی مرتبه سیستم بالا باشد این کار بسیار مشکل و وقت‌گیر است. اگر هر کدام از ماتریس‌های موجود در (۱۵.۲) متغیر با زمان باشند، در آن حالت به‌طورکلی باید برای برآورد  $\phi(t_f, t)$  از روش‌های عددی استفاده نمود.

راه دیگری نیز وجود دارد و آن این است که می‌توان نشان داد که ماتریس  $K$  در معادله دیفرانسیل ماتریسی زیر با شرایط حدی  $K(t_f) = H$  صدق می‌کند.

$$\dot{K} = -K(t)A(t) - A^T(t)K(t) - Q(t) + K(t)B(t)R^{-1}(t)B^T(t)K(t). \quad (20.2)$$

این معادله دیفرانسیل ماتریسی از نوع ریکاتی می‌باشد. معادله (۲۰.۲) را معادله ریکاتی می‌نامیم. چون  $K$  یک ماتریس  $n \times n$  است، معادله (۲۰.۲) سیستمی از  $n^2$  معادله دیفرانسیل مرتبه اول می‌باشد. می‌توان نشان داد که  $K$  متقارن است بنابراین به جای  $n^2$  باید  $\frac{(n)(n+1)}{2}$  معادله دیفرانسیل مرتبه اول حل شود. این انتگرالگیری از زمان  $t = t_f$  شروع شده و در جهت عکس پیش می‌رود تا به زمان  $t = t_0$  برسد،  $K(t)$  ذخیره شده و ماتریس ضریب تقویت بازخورد از معادله (۱۹.۲) تعیین می‌شود. یافتن جواب

<sup>۲۸</sup>Kalman<sup>۲۹</sup>Feedback gain matrix



شکل ۱۰۲: دستگاه و کنترل کننده بازخورد بهینه برای مساله تنظیم کننده خطی

دقیق مساله تنظیم کننده در مسائل با بعد بزرگ کاری دشوار و با خطای محاسباتی بالایی همراه است. به عنوان مثال برای مسائل با بعد بالا محاسبه  $R^{-1}$  بسیار دشوار است. بنابراین ناگزیریم با استفاده از روش های عددی به حل مساله (۱۱.۲) با تابع هدف (۱۰.۲) بپردازیم. یکی از کاراترین روش ها استفاده از روش شبکه های عصبی است که در فصل سوم توضیح کاملی در مورد آن ارائه می دهیم. اینک به حل یک مثال عددی به روش تشریحی می پردازیم.

## ۷.۲ مثال عددی

مثال ۱۰.۷.۲. قانون کنترل بهینه را برای سیستم

$$\dot{x}(t) = ax(t) + u(t),$$

چنان بیابید که تابع هدف

$$J = \frac{1}{2} H x^2(T) + \int_0^T \frac{1}{2} u^2(t) dt.$$

را حداقل کند. مقادیر موقعیت ها و کنترل های قابل قبول محدود نمی باشند، زمان نهایی  $T$  مشخص می باشد،  $H > 0$  و  $x(T)$  آزاد یا غیرمعیّن است. از معادله (۱۵.۲) داریم:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{p}(t) \end{bmatrix} = \begin{bmatrix} a & -\lambda \\ 0 & -a \end{bmatrix} \begin{bmatrix} x(t) \\ p(t) \end{bmatrix},$$

که دارای ماتریس انتقال زیر می باشد:

$$\phi(t) = \begin{bmatrix} e^{at} & \frac{1}{a} e^{-at} - \frac{1}{a} e^{at} \\ 0 & e^{-at} \end{bmatrix},$$

بنابراین از معادله (۱۷.۲) و (۱۸.۲) داریم:

$$K(t) = \left[ e^{-a(T-t)} - \frac{H}{a} [e^{-a(T-t)} - e^{a(T-t)}] \right]^{-1} [H e^{a(T-t)}].$$

و از معادله (۱۹.۲) قانون کنترل بهینه عبارتست از:

$$u(t) = -\gamma K(t)x(t),$$

که در آن:

$$H = 5, \quad a = -0.2, \quad x(0) = 5, \quad T = 15. \quad (21.2)$$



# فصل ۳

## شرایط بهینگی

### ۱.۳ مقدمه

در این فصل می‌خواهیم یک روش جدید بر اساس توانایی شبکه‌های عصبی مصنوعی در تقریب تابع ارائه دهیم تا جواب مساله کنترل بهینه را بدست آوریم. برای انجام این کار سعی می‌کنیم شرایط هامیلتونین را بر پایه‌ی اصل مینیمم پونتریاگین (PMP) تقریب بزیم، سپس تابع خطا را که شامل همه‌ی شرایط پونتریاگین است را معرفی می‌کنیم. در تابع خطا پیشنهاد شده، از جواب‌های آزمایشی برای تابع مسیر، تابع کنترل و ضرایب لاگرانژ استفاده می‌کنیم. این جواب‌های آزمایشی بوسیله‌ی نرون‌ها ساخته می‌شوند. بنابراین، تابع خطا را که شامل وزن همه‌ی جواب‌های آزمایشی است را مینیمم می‌کنیم. با جایگذاری مقادیر بهینه وزن‌ها در جواب‌های آزمایشی، تابع بهینه مسیر، تابع بهینه کنترل و همچنین ضرایب بهینه لاگرانژ را بدست می‌آوریم.

### ۲.۳ تعاریف اولیه

در این بخش مدل زیر از مساله کنترل بهینه را در نظر می‌گیریم

$$\text{minimize } \int_{t_0}^{t_f} f_0(x(t), u(t), t) dt, \quad (1.3)$$

subject to

$$\dot{x} = g(x(t), u(t), t), \quad (2.3)$$

$$x(t_0) = x_0. \quad (3.3)$$

که در آن  $x(t) \in \mathbb{R}^n$  متغیر حالت،  $u(t) \in \mathbb{R}^m$  متغیر کنترل و  $t \in [t_0, t_f]$ . فرض می‌کنیم تابع زیر انتگرال  $f_0$  دارای مشتق جزئی پیوسته مرتبه‌ی اول و دوم نسبت به تمام متغیرها باشد. همچنین فرض می‌کنیم که  $t_0$  و  $t_f$  ثابت و  $g$  تابع لیپ شیتس پیوسته روی مجموعه‌ی  $\Omega \subset \mathbb{R}^{n+m+1}$  باشد. بنا به مساله

(۳.۳)-(۱.۳) می‌توانیم تابع همیلتونین را بصورت زیر بسازیم:

$$H(x(t), u(t), p(t), t) = f_0(x(t), u(t), t) + p(t) \cdot g(x(t), u(t), t),$$

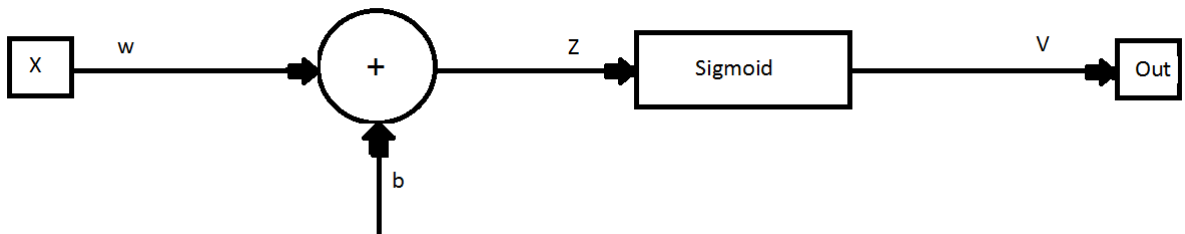
که در آن  $p \in \mathbb{R}^m$  بردار الحاقی است. فرض کنیم متغیرهای بهینه حالت، الحاقی و کنترل را به ترتیب با  $x^*(t), p^*(t)$  و  $u^*(t)$  نمایش دهیم. شرط لازم برای اینکه  $u^*(t)$  مینیمم کننده تابعی هدف در (۳.۳)-(۱.۳) برای هر  $t \in [t_0, t_f]$  و برای هر کنترل‌های قابل قبول باشد این است که:

$$H(x^*(t), u^*(t), p^*(t), t) \leq H(x^*(t), u(t), p^*(t), t) \quad (۴.۳)$$

معادله‌ی (۴.۳) نشان می‌دهد که کنترل بهینه همیلتونین باید مینیمم شود که آن را PMP (اصل مینیمم پونتریاگین) می‌گویند [۲۵]. همچنین PMP شرط لازم برای بهینگی را فراهم می‌کند. PMP نشان می‌دهد که اگر  $x(t), p(t)$  و  $u(t)$  به ترتیب مقادیر بهینه حالت، الحاقی و کنترل باشند، باید در شرایط زیر صدق کنند:

$$\begin{cases} \frac{\partial H(x, u, t, p)}{\partial x} = -\dot{p}(t), \\ \frac{\partial H(x, u, t, p)}{\partial u} = \dot{x}(t), \\ \frac{\partial H(x, u, t, p)}{\partial p} = 0. \end{cases} \quad (۵.۳)$$

بوسیله‌ی جایگذاری توابع مجهول  $f_0$  و  $g$  در همیلتونین، معادله‌ی (۵.۳) یک سیستم معادله دیفرانسیل معمولی می‌دهد که می‌توان آن را با روش‌های عددی یا دیگر روش‌های موجود حل کرد. در برخی موارد معادله‌ی (۵.۳) یک سیستم معادله دیفرانسیل معمولی را فراهم می‌کند که به آسانی می‌توان آن را حل کرد اما در بیشتر موارد (مخصوصاً در مسائل کاربردی)، سیستم به راحتی حل نمی‌شود و باید روش‌های تقریبی بکار گرفته شود. در بخش بعد، سعی می‌کنیم توانایی شبکه عصبی را در تخمین تابع برای حل معادله‌ی (۵.۳) بکار ببریم. نرون‌های اولیه روی پرسپترون‌ها را می‌توان در شکل (۱.۳) مشاهده نمود. این موجب می‌شود ما بتوانیم از چند لایه پرسپترون برای تخمین تابع غیر خطی با دقت دلخواه استفاده کنیم [۴۰]. در اینجا  $W$  بردار وزن لایه‌های داخلی،  $b$  بردار شامل وزن‌های بایاس و  $V$  وزن‌های لایه‌ی خروجی است.



شکل ۱.۳: نرون‌های اولیه

می‌توان مشاهده کرد که خروجی بوسیله‌ی فرمول زیر محاسبه می‌شود:

$$\begin{cases} \text{output} = \sum_{i=1}^I v_i \sigma(z_i), \\ z_i = w_i x + b_i, \end{cases} \quad (۶.۳)$$

که در اینجا  $I$  تعداد سیگموئیدهای واحد است. تابع فعال سازی  $\sigma$ ، که یک تابع سیگموئید است بصورت زیر در نظر گرفته می‌شود:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (۷.۳)$$

بنابر قضیه کولومگوروف، می‌توانیم هر تابع پیوسته را با پرسپترون چندلایه تقریب بزنیم. بوسیله‌ی این قضیه می‌توانیم از هر شبکه عصبی دلخواه در تخمین تابع استفاده کنیم. تخمین تابع حالت، الحاقی و کنترل برای مساله کنترل بهینه (۳.۳)-(۱.۳) با استفاده از این اصل در بخش بعد بحث خواهد شد.

### ۳.۳ ساختار شبکه عصبی

در این بخش سعی می‌کنیم طرح تخمینی برای حل معادله (۵.۳) پیشنهاد کنیم [۵۸]. برای هر تابع، سه شبکه عصبی در نظر می‌گیریم: حالت (شبکه‌ی عصبی متاظر آن  $n_x$ )، الحاقی (شبکه‌ی عصبی متاظر آن  $n_p$ ) و کنترل (شبکه‌ی عصبی متاظر آن  $n_u$ ). هر شبکه عصبی شامل پارامترهای قابل تنظیم است که در شکل (۱.۳) می‌توان آن‌ها را مشاهده کرد. ساختار مدل‌های شبکه عصبی باید در شرایط اولیه و مرزی صدق کنند. مدل‌های شبکه عصبی می‌توانند بصورت زیر پیشنهاد شوند:

$$\begin{cases} n_x = \sum_{i=1}^I v_x^i \bar{\sigma}(z_x^i) & ; z_x^i = w_x^i t + b_x^i \\ n_p = \sum_{i=1}^I v_p^i \bar{\sigma}(z_p^i) & ; z_p^i = w_p^i t + b_p^i \\ n_u = \sum_{i=1}^I v_u^i \bar{\sigma}(z_u^i) & ; z_u^i = w_u^i t + b_u^i \end{cases} \quad (۸.۳)$$

برای  $i = 1, 2, \dots, I$  که  $I$  تعداد نرون‌هایی است که می‌تواند برای هر شبکه عصبی متفاوت باشد و

$$\bar{\sigma} = (t_f - t_0)\sigma(z) + t_0.$$

حال می‌خواهیم از شبکه‌های عصبی (۸.۳) استفاده کنیم و جواب‌های آزمایشی را تعریف کنیم. جواب‌های آزمایشی (برای توابع حالت، الحاقی و کنترل) شامل شبکه‌های عصبی است که در شرایط اولیه و مرزی صدق می‌کنند، بنابراین آن‌ها می‌توانند بوسیله‌ی ساختار زیر تعریف شوند:

$$\begin{cases} x_T = x_0 + (t - t_0)n_x \\ p_T = n_p \\ u_T = n_u \end{cases} \quad (۹.۳)$$

به راحتی می‌توان بررسی کرد که  $x_T$  در شرایط اولیه صدق می‌کند ( $x_T(t_0) = x_0$ ). متذکر می‌شویم برای نقاط انتهایی آزاد داریم  $p(\cdot) = 0$ . برای مثال، اگر  $x(t_f)$  آزاد باشد، باید داشته باشیم  $p(t_f) = 0$  و بنابراین برای  $p_T$  در (۹.۳) تعریف می‌کنیم:

$$p_T = (t - t_f)n_p.$$

برای دیگر شرایط اولیه (یا مرزی) می‌توانیم تابع آزمایشی تخمینی را بسازیم.

بوسیله‌ی جایگذاری جواب آزمایشی در تابع هامیلتونین، هامیلتونین آزمایشی  $H_T$  را تعریف کنیم که همان تابع هامیلتونین معمولی  $H$  با جایگذاری توابع  $x, p$  و  $u$  بجای صورت آزمایشی متناظر آن‌ها  $(x_T, p_T, u_T)$  بصورت زیر حاصل می‌شود:

$$H_T(x_T(t), u_T(t), p_T(t), t) = f_0(x_T(t), u_T(t), t) + p_T(t).g(x_T(t), u_T(t), t)$$

بنابراین تابع هامیلتونین آزمایشی شامل وزن‌های شبکه‌های عصبی است. چون جواب آزمایشی (۹.۳) باید در شرایط (۵.۳) صدق کند، با جایگذاری آن‌ها در (۵.۳) داریم:

$$\begin{cases} \frac{\partial H_T}{\partial x_T} + \dot{p}_T = 0 \\ \frac{\partial H_T}{\partial p_T} - \dot{x}_T = 0 \\ \frac{\partial H_T}{\partial u_T} = 0 \end{cases} \quad (10.3)$$

برای حل سیستم (۱۰.۳)، ۳ تابع خطا متناظر با هر معادله تعریف می‌کنیم:

$$\begin{cases} E_1(t, y) = \left[ \frac{\partial H_T}{\partial x_T} + \dot{p}_T \right]^2 \\ E_2(t, y) = \left[ \frac{\partial H_T}{\partial p_T} - \dot{x}_T \right]^2 \\ E_3(t, y) = \left[ \frac{\partial H_T}{\partial u_T} \right]^2 \end{cases} \quad (11.3)$$

و در نهایت تابع خطا بصورت زیر است:

$$E(t, y) = E_1(t, y) + E_2(t, y) + E_3(t, y)$$

که در آن  $y$  یک بردار شامل تمام وزن‌های ۳ شبکه عصبی (۸.۳) است. به علاوه،  $y$  شامل تمام وزن‌های  $w_x, w_p, w_u, b_x, b_p, b_u, v_x, v_p, v_u$  است. حال بجای حل معادله (۱۰.۳) بازه  $[t_0, t_f]$  را با  $m$  نقطه  $k = 1, 2, \dots, m, t_k$  گسسته سازی می‌کنیم و مساله بهینه سازی نامقید زیر را حل می‌کنیم.

$$\underset{y}{\text{minimize}} E(y) = \sum_{k=1}^m \{E_1(t_k, y) + E_2(t_k, y) + E_3(t_k, y)\} \quad (12.3)$$



لم ۱.۳.۳. اگر  $y^* = (w_x^*, w_p^*, w_u^*, b_x^*, b_p^*, b_u^*, v_x^*, v_p^*, v_u^*)^T$  در دستگاه زیر صدق کند:

$$\eta(y) = \begin{bmatrix} E_1(t_1, y) \\ E_1(t_2, y) \\ \vdots \\ E_1(t_m, y) \\ E_2(t_1, y) \\ E_2(t_2, y) \\ \vdots \\ E_2(t_m, y) \\ E_3(t_1, y) \\ E_3(t_2, y) \\ \vdots \\ E_3(t_m, y) \end{bmatrix} = 0, \quad (13.3)$$

در اینصورت  $y^*$  جواب بهینه (۱۲.۳) است.

برهان. فرض کنیم  $\eta(y^*) = 0$  در اینصورت  $E_i(t_k, y^*) = 0$  برای  $i = 1, 2, 3$  و  $k = 1, 2, \dots, m$ . چون در (۱۲.۳)  $E(y) \geq 0$ ، بنابراین  $y^*$  جواب بهینه (۱۲.۳) است. □

بنابر لم (۱.۳.۳) می‌توان مشاهده کرد که مساله بهینه سازی (۱۲.۳) با مساله زیر هم ارز است

$$\text{minimize}_y E(y) = \frac{1}{2} \|\eta(y)\|^2. \quad (14.3)$$

برای حل (۱۴.۳) که یک مساله بهینه سازی نامقید است، می‌توان از الگوریتم‌های بهینه سازی مانند تندترین کاهش، نیوتن، شبه نیوتن، گرادیان مزدوج و غیره استفاده کرد [۲۰]–[۳۳]. همچنین می‌توانیم از الگوریتم‌های ابتکاری مانند الگوریتم ژنتیک، بهینه سازی گروه پرندگان، الگوریتم مورچگان و غیره استفاده کنیم [۳۹].

بعد از پایان مراحل بهینه سازی می‌توانیم مقادیر بهینه وزن‌های  $y$  (که شامل وزن لایه‌های داخلی و خارجی و بردار بایاس است) در معادله‌ی (۹.۳) جایگذاری کنیم و ساختارهای توابع حالت، الحاقی و کنترل را محاسبه کنیم.

مزیت اصلی این روش این است که اجرای الگوریتم خیلی پیچیده نیست، ما می‌توانیم از لایه‌های پنهانی بیشتری یا نقاط آزمایشی بیشتری روی بازه‌ی  $[t_o, t_f]$  استفاده کنیم تا دقت تخمین بیشتری بدست آوریم. در نهایت، بردار حالت، الحاقی و کنترل بصورت تابعی از زمان  $t$  مطرح می‌شود. بنابراین می‌توانیم جواب را در هر نقطه‌ی دلخواه روی بازه  $[t_o, t_f]$  محاسبه کنیم. همچنین تابع کنترل و حالت مشتق پذیر هستند که می‌تواند در کاربرد مفید باشد.

### ۴.۳ مثال‌های عددی

در این بخش، سعی می‌کنیم الگوریتم پیشنهادی را با حل چند مثال نشان دهیم. در همه مثال‌ها ۵ پارامتر برای هر ورودی، خروجی و وزن بایاس استفاده می‌کنیم. در همه مثال‌ها از نرم افزار GAMS برای شبیه‌سازی استفاده کرده‌ایم.

مثال ۱.۴.۳. مساله بهینه سازی زیر را در نظر بگیرید:

$$\text{minimize } \frac{H}{4} x^2(T) + \int_0^T \frac{1}{4} u^2(t) dt,$$

subject to

$$\dot{x}(t) = -ax(t) + u(t),$$

$$x(0) = 5, x(T) = \text{is free}.$$

با فرض  $T = 15$ ,  $a = -0.2$ ,  $H = 5$ , هامیلتونین را به صورت زیر می‌نویسیم،

$$H = \frac{1}{4} u^2 + p(-0.2x + u).$$

از شرایط بهینگی (۵.۳) داریم:

$$\begin{cases} -0.2p = -\dot{p}, \\ \frac{1}{2}u + p = 0, \\ \dot{x} = -0.2x + u. \end{cases}$$

و چون  $x$  در  $T = 15$  آزاد است داریم:

$$\frac{\partial H(T)}{\partial x(T)} = p(T) \implies 5x(15) = p(15)$$

حال با حل دستگاه معادله دیفرانسیل فوق، جواب دقیق متغیر حالت و کنترل به صورت زیر است:

$$\begin{cases} x(t) = B \exp\left(\frac{-t}{5}\right) - 5A \exp\left(\frac{t}{5}\right), \\ u(t) = -2A \exp\left(\frac{t}{5}\right). \end{cases}$$

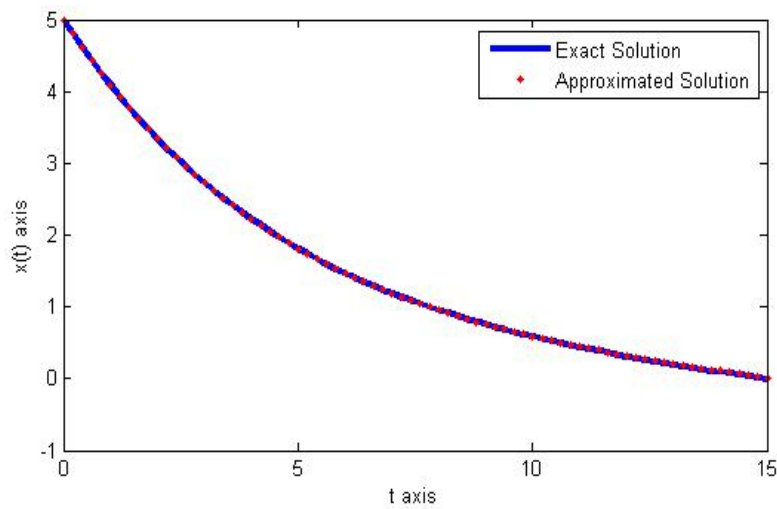
و با استفاده از شرایط اولیه و انتهایی پارامترهای  $A$  و  $B$  به صورت زیر محاسبه می‌شوند.

$$A = \frac{25}{26 \exp(6) - 25}, \quad B = \frac{26A \exp(6)}{5}.$$

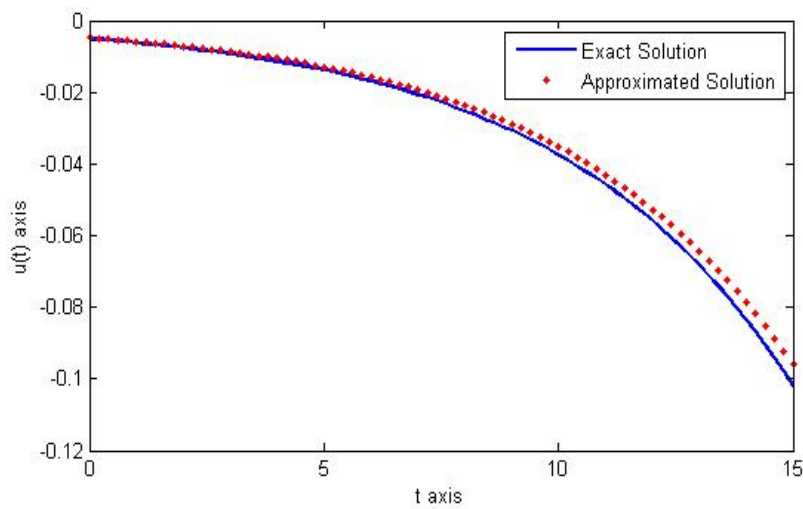
با در نظر گرفتن شرایط اولیه و انتهایی، می‌توانیم جواب‌های آزمایشی را به صورت زیر انتخاب کنیم:

$$\begin{cases} x_T = 5 + tn_x, \\ p_T = (t - 15)n_p + 5x(15), \\ u_T = n_u. \end{cases}$$

می‌توانیم جواب‌های تقریبی برای  $x(t)$  و  $u(t)$  را به ترتیب در شکل‌های (۲.۳) و (۳.۳) مشاهده کنیم.



شکل ۲.۳: تابع حالت برای مثال (۱.۴.۳)



شکل ۳.۳: تابع کنترل برای مثال (۱.۴.۳)

مثال ۲.۴.۳. مساله بهینه سازی زیر را در نظر بگیرید:

$$\text{minimize } \frac{1}{2} \int_0^1 x^2(t) + u^2(t) dt,$$

subject to

$$\dot{x}(t) = -x(t) + u(t),$$

$$x(0) = 1, x(1) = \text{is free}.$$

هامیلتونین را به صورت زیر می‌نویسیم:

$$H = \frac{1}{2}x^2 + \frac{1}{2}u^2 + p(-x + u)$$

و از شرایط بهینگی (۵.۳) داریم:

$$\begin{cases} x - p = -\dot{p}, \\ u + p = 0, \\ \dot{x} = -x + u. \end{cases}$$

چون  $x = 1$  در  $t = 0$  آزاد است داریم  $p(0) = 0$ . حال با حل دستگاه معادله دیفرانسیل فوق، جواب دقیق متغیر حالت و کنترل به صورت زیر است:

$$\begin{cases} x(t) = A\exp(\sqrt{2}t) + (1 - A)\exp(-\sqrt{2}t), \\ u(t) = A(\sqrt{2} + 1)\exp(\sqrt{2}t) - (1 - A)(\sqrt{2} - 1)\exp(-\sqrt{2}t). \end{cases}$$

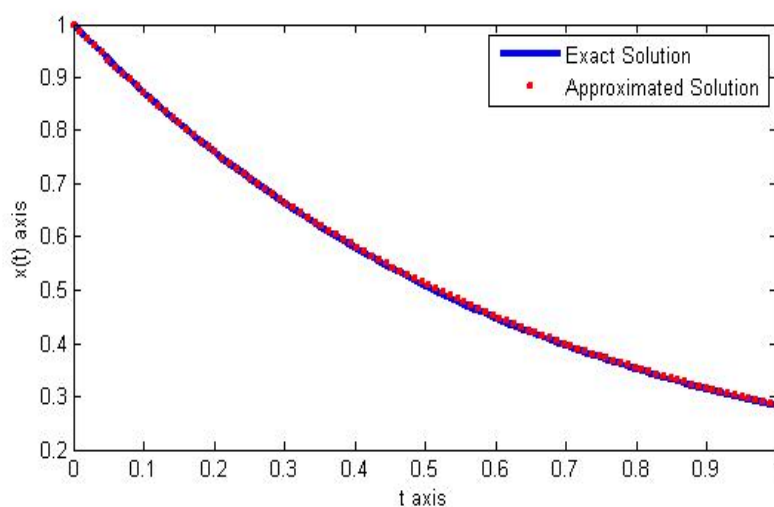
با استفاده از شرایط اولیه و انتهایی پارامتر  $A$  به صورت زیر محاسبه می‌شود:

$$A = \frac{2\sqrt{2} - 3}{2\sqrt{2} - 3 - \exp(2\sqrt{2})}.$$

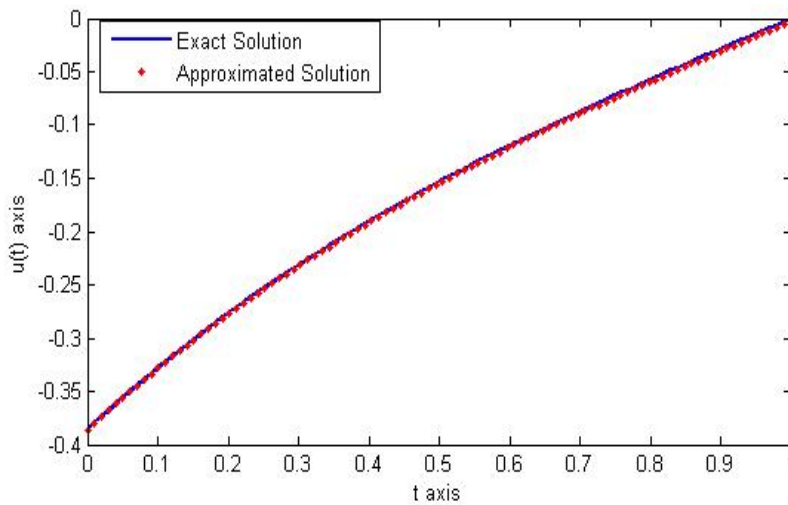
با در نظر گرفتن شرایط اولیه و انتهایی، می‌توانیم جواب‌های آزمایشی را به صورت زیر انتخاب کنیم:

$$\begin{cases} x_T = 1 + tn_x, \\ p_T = (t - 1)n_p, \\ u_T = n_u. \end{cases}$$

می‌توانیم جواب‌های تقریبی برای  $x(t)$  و  $u(t)$  را به ترتیب در شکل‌های (۴.۳) و (۵.۳) مشاهده کنیم.



شکل ۴.۳: تابع حالت برای مثال (۲.۴.۳)



شکل ۵.۳: تابع کنترل برای مثال (۲.۴.۳)

مثال ۳.۴.۳. مساله بهینه سازی زیر را در نظر بگیرید:

$$\text{minimize } x^2(1) + \frac{1}{2} \int_0^1 u^2(t) dt,$$

subject to

$$\dot{x}(t) = x(t) + u(t),$$

$$x(0) = 1, x(1) = \text{is free}.$$

با استفاده از (۱.۳) هامیلتونین را به صورت زیر می‌نویسیم

$$H = \frac{1}{2} u^2 + p(x + u)$$

از شرایط بهینگی (۵.۳) داریم:

$$\begin{cases} p = -\dot{p}, \\ u + p = 0, \\ \dot{x} = x + u, \end{cases}$$

و چون  $x$  در  $t = 1$  آزاد است داریم:

$$\frac{\partial H(t)}{\partial x(t)} = p(t) \implies 2x(1) = p(1)$$

حال با حل دستگاه معادله دیفرانسیل فوق، جواب دقیق متغیر حالت و کنترل به صورت زیر است:

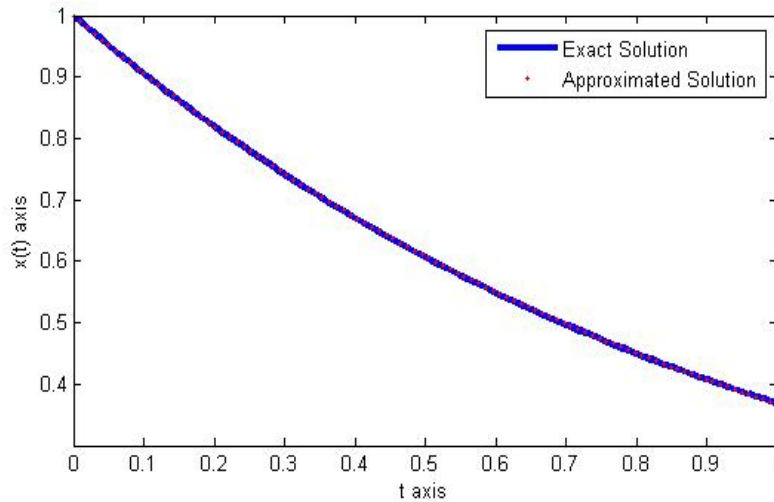
$$\begin{cases} x(t) = \exp(-t), \\ u(t) = -A \exp(-t). \end{cases}$$

با استفاده از شرایط اولیه و انتهایی داریم  $A = 2$ .

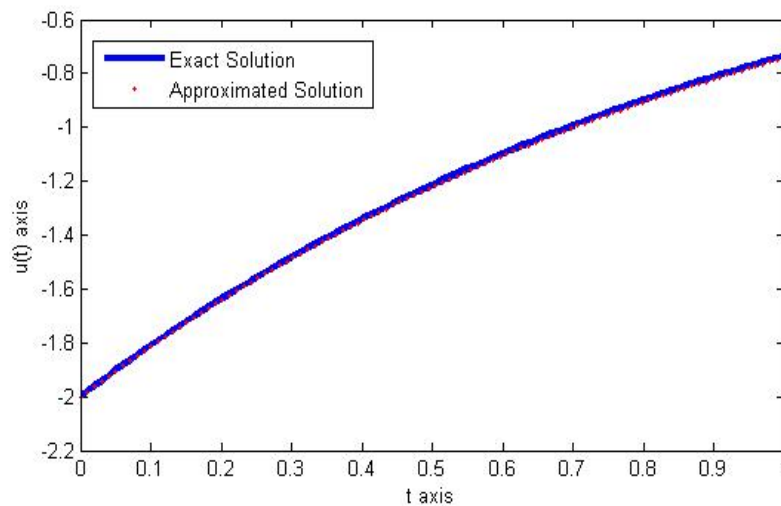
با در نظر گرفتن شرایط اولیه و انتهایی، می‌توانیم جواب‌های آزمایشی را به صورت زیر انتخاب کنیم:

$$\begin{cases} x_T = 1 + tn_x, \\ p_T = (t - 1)n_p, \\ u_T = n_u. \end{cases}$$

می‌توانیم جواب‌های تقریبی برای  $x(t)$  و  $u(t)$  را به ترتیب در شکل‌های (۶.۳) و (۷.۳) مشاهده کنیم.



شکل ۶.۳: تابع حالت برای مثال (۳.۴.۳)



شکل ۷.۳: تابع کنترل برای مثال (۳.۴.۳)

مثال ۴.۴.۳. [۵۸] مساله بهینه سازی زیر را در نظر بگیرید:

$$\text{minimize } \int_0^1 u^2(t) dt,$$

subject to

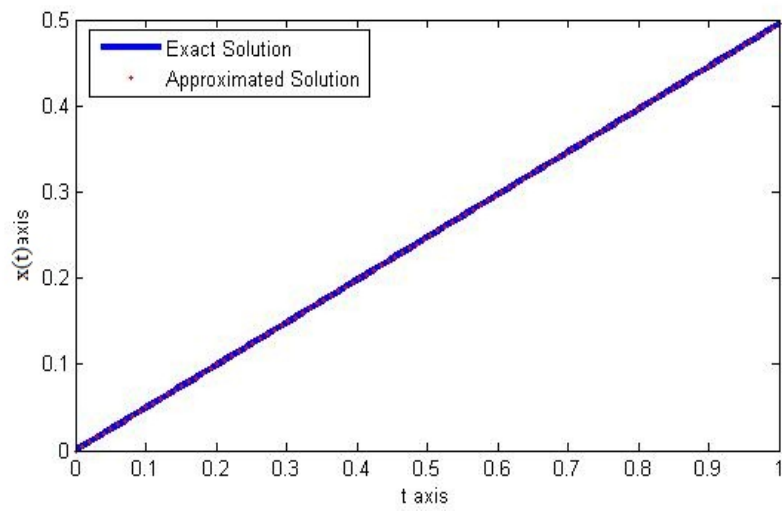
$$\dot{x}(t) = 0.5x^2(t) \sin(x(t)) + u(t),$$

$$x(0) = 0, x(1) = 0.5.$$

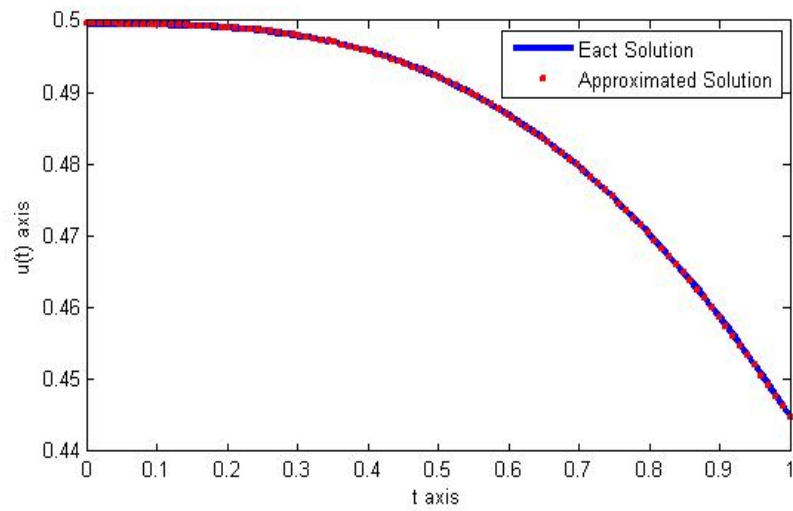
با استفاده از شرایط اولیه و انتهایی، این سیستم را می‌توان بوسیله‌ی روش‌های عددی (مانند روش اویلر) حل نمود و نتایج را با نتایج به دست آمده توسط شبکه عصبی مقایسه کرد. با استفاده از شرایط اولیه و انتهایی، می‌توانیم جواب‌های آزمایشی را به صورت زیر انتخاب کنیم:

$$\begin{cases} x_T = t(1-t)n_x + 0.5t, \\ p_T = n_p, \\ u_T = n_u. \end{cases}$$

می‌توانیم جواب‌های تقریبی برای  $x(t)$  و  $u(t)$  را به ترتیب در شکل‌های (۸.۳) و (۹.۳) مشاهده کنیم.



شکل ۸.۳: تابع حالت برای مثال (۴.۴.۳)



شکل ۹.۳: تابع کنترل برای مثال (۴.۴.۳)



# فصل ۴

## مسائل کنترل بهینه با قیدهای نامساوی

### ۱.۴ مقدمه

در این فصل، رده‌ای از مسائل کنترل بهینه غیر خطی با قیدهای نامساوی را در نظر می‌گیریم. با استفاده از شرایط بهینگی کاروش-کان-تاکر<sup>۱</sup> برای مسائل بهینه سازی غیر خطی و بوسیله‌ی ساختن تابع خطا، یک مساله بهینه سازی نامقید را تعریف می‌کنیم. در مساله بهینه سازی از جواب‌های آزمایشی برای توابع حالت، کنترل و ضرایب لاگرانژ استفاده می‌کنیم که در آن این جواب‌های آزمایشی بوسیله‌ی پرسپترون دو لایه<sup>۲</sup> ساخته می‌شوند. سپس تابع خطا که در آن وزن‌ها و بایاس‌ها به نرون‌ها وابسته هستند، را مینیمم می‌کنیم. با جایگزینی مقادیر بهینه وزن‌ها و بایاس‌ها در جواب‌های آزمایشی، جواب بهینه مساله اصلی را به دست می‌آوریم. به عنوان یک مثال کاربردی، روش پیشنهاد شده را برای مساله بازوی مکانیکی ربات بکار می‌بریم.

### ۲.۴ بیان مساله

در این بخش یک تکنیک تحلیلی برای حل مسائل بهینه سازی دینامیک شامل قیدهای نامساوی ارائه می‌دهیم [۵۹]. مساله بهینه سازی زیر را در نظر بگیرید:

$$\text{minimize } J(\bar{x}, \bar{u}) = \Phi(t, \bar{x})|_{t_f} + \int_{t_0}^{t_f} L(t, \bar{x}, \bar{u}) dt, \quad (1.4)$$

subject to,

$$\bar{x}_i = \bar{f}_i(t, \bar{x}, \bar{u}), \quad i = 1, 2, \dots, n, \quad (2.4)$$

$$c_l(\bar{x}, \bar{u}, t) \leq 0, \quad l = 1, 2, \dots, p, \quad (3.4)$$

$$\bar{x}(t_0) = \bar{x}_0. \quad (4.4)$$

<sup>۱</sup>Karush-Kuhn-Tucker

<sup>۲</sup>Tow-Layered-Perceptron

که در آن  $\bar{x}(t) \in \mathbb{R}^n$  و  $t_0$  و  $t_f$  ثابت هستند.  $p$  قید نامساوی  $c_l, l = 1, 2, \dots, p$  شامل قیدهای نامساوی متغیر کنترل (که در آن متغیر کنترل صریحا ظاهر می‌شود) و همچنین قیدهای نامساوی متغیر حالت (که در آن متغیر کنترل ظاهر نمی‌شود) است. با اضافه کردن معادله‌های (۲.۴) و (۳.۴) در تابعی (۱.۴) با ضرایب لاگرانژ  $\bar{\lambda} \in \mathbb{R}^n$  و  $\bar{\mu} \in \mathbb{R}^p$  داریم:

$$J' = \Phi + \int_{t_0}^{t_f} \left( L + \sum_{i=1}^n \lambda_i (f_i - \dot{x}_i) + \sum_{l=1}^p (\mu_l c_l) \right)$$

با قراردادن  $\delta J' = 0$ ، شرایط لازم مرتبه‌ی اول به صورت زیر به دست می‌آید:

$$\dot{x}_i = f_i, \quad i = 1, 2, \dots, n, \quad (5.4)$$

$$\dot{\lambda}_i = -\frac{\partial L}{\partial x_i} - \sum_{i=1}^n \frac{\partial f_i}{\partial x_i} \lambda_i - \sum_{l=1}^p \frac{\partial c_l}{\partial x_i} \mu_l, \quad i = 1, 2, \dots, n, \quad (6.4)$$

$$0 = \frac{\partial L}{\partial u_s} + \sum_{i=1}^n \frac{\partial f_i}{\partial u_s} \lambda_i + \sum_{l=1}^p \frac{\partial c_l}{\partial u_s} \mu_l, \quad s = 1, 2, \dots, m, \quad (7.4)$$

$$0 = \mu_l c_l, \quad l = 1, 2, \dots, p, \quad (8.4)$$

$$0 \leq \mu_l, \quad l = 1, 2, \dots, p, \quad (9.4)$$

$$0 \geq c_l, \quad l = 1, 2, \dots, p, \quad (10.4)$$

$$x_i(t_0) = x_{0i}, \quad i = 1, 2, \dots, n, \quad (11.4)$$

$$\lambda_i(t_f) = \frac{\partial \Phi}{\partial x_i} \Big|_{t_f}, \quad i = 1, 2, \dots, n. \quad (12.4)$$

معادلات (۱۲.۴)–(۵.۴)، نشان دهنده‌ی مساله‌ی مقدار مرزی<sup>۳</sup> شامل معادلات جبری و دیفرانسیل<sup>۴</sup> است، که در آن (۵.۴) و (۶.۴) معادلات دیفرانسیل، (۱۰.۴)–(۷.۴) معادلات جبری و (۱۱.۴) و (۱۲.۴) شرایط مرزی هستند.

می‌توانیم بین جواب مساله مکمل غیرخطی<sup>۵</sup>  $(NCP)$ –(۱۰.۴)–(۸.۴) و معادله‌ی هم ارز آن با استفاده از توابع  $NCP$ ، ارتباط برقرار کنیم. رده‌ای از توابع  $NCP$  که در زیر تعریف شده در ساختن توابع شایسته<sup>۶</sup> استفاده می‌شود.

تعریف ۱۰.۲.۴. تابع  $\Phi: \mathbb{R}^2 \rightarrow \mathbb{R}$  تابع  $NCP$ –نامیده می‌شود اگر در شرط زیر صدق کند

$$\Phi(a, b) = 0 \iff a \geq 0, \quad b \geq 0, \quad ab = 0.$$

• یک تابع  $NCP$  عمومی، تابع فیشر–بورمیستر<sup>۷</sup>  $(FB)$  نامیده می‌شوند که نیمه هموار قوی<sup>۸</sup> [۲۶، ۶۱] می‌باشند و به صورت زیر تعریف می‌شوند

$$\Phi_{FB}(a, b) = (a + b) - \sqrt{a^2 + b^2}.$$

<sup>۳</sup>Boundary Value Problem

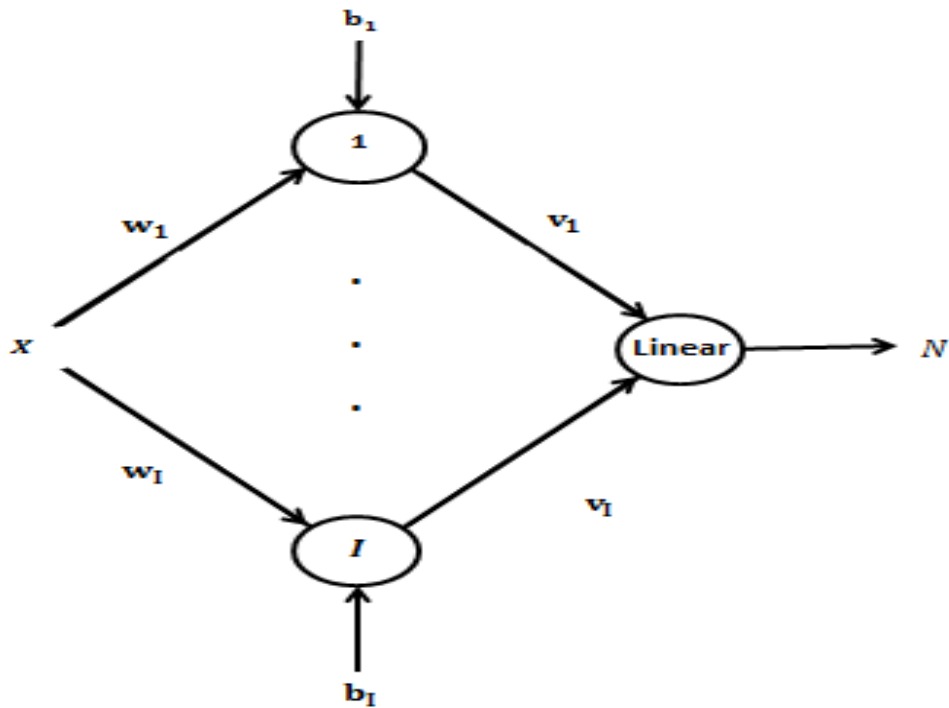
<sup>۶</sup>Metric

<sup>۴</sup>Differential and Algebraic Equation

<sup>۷</sup>Fischer-Burmister

<sup>۵</sup>Nonlinear Complementarity Problem

<sup>۸</sup>Strongly Semismooth



شکل ۱.۴: پرسپترون دولایه با یک ورودی و یک خروجی

تابع  $FB$  اغتشاش یافته به صورت زیر داده می‌شود:

$$\Phi_{FB}^\epsilon(a, b) = (a + b) - \sqrt{a^2 + b^2 + \epsilon}, \quad \epsilon \rightarrow 0_+.$$

مهمترین ویژگی  $\Phi_{FB}^\epsilon$  در گزاره زیر می‌تواند بیان شود.

گزاره ۲.۲.۴. [۲۹]

$$\Phi_{FB}^\epsilon(a, b) = 0 \Leftrightarrow a > b, b > 0, ab = \frac{\epsilon}{4} \quad \forall \epsilon \in \mathbb{R}.$$

گزاره بالا واضح است و اثبات آن حذف شده است. یادآور می‌شویم که  $\Phi_{FB}^\epsilon(a, b)$  نسبت به  $a, b$  برای هر  $\epsilon > 0$  هموار است.

با استفاده از تابع  $FB$  معرفی شده، می‌توانیم مساله مکمل غیرخطی (۱۰.۴)-(۸.۴) را به قید مساوی بصورت زیر تبدیل کنیم:

$$\Phi_{FB}^\epsilon(\mu_l, c_l) = 0, \quad \epsilon \rightarrow 0_+, \quad l = 1, \dots, p. \quad (13.4)$$

### ۳.۴ ساختار شبکه عصبی

واضح است که می‌توانیم خروجی شبکه عصبی دولایه شکل (۱.۴) را به صورت زیر محاسبه کنیم:

$$\begin{cases} \text{output} = \sum_{i=1}^I v_i \sigma(z_i), \\ z_i = w_i x + b_i, \end{cases} \quad (14.4)$$

که در آن  $I$  تعداد سیگموئیدهای واحد،  $w_i$  پارامتر وزن از ورودی به  $i$ -امین واحد پنهان شده،  $v_i$ ،  $i$ -امین پارامتر وزن از واحد پنهان شده  $i$  به لایه خروجی،  $b_i$ ،  $i$ -امین بایاس برای  $i$ -امین واحد در لایه پنهان شده،  $z_i$  ورودی  $i$ -امین واحد پنهان شده و  $\sigma$  تابع سیگموئید دلخواه است. تابع تبدیل  $\sigma$  عموماً یک تابع یک بعدی غیرخطی یکنوا (که تابع فعال سازی نام دارد) است، که می‌تواند به آسانی محاسبه شود. در مثال‌های عددی این بخش، تابع سیگموئید زیر در نظر گرفته شده است:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \quad (15.4)$$

واضح است که هر تابع به اندازه‌ی کافی هموار را می‌توانیم روی مجموعه‌های فشرده با استفاده از شبکه عصبی دولایه با وزن‌های مناسب، تقریب بزنیم (برای اطلاعات بیشتر به [۴۱، ۳۰، ۴۲] مراجعه کنید). این نتایج نشان می‌دهد که هر تابع پیوسته را می‌توان با استفاده از ترکیبی از توابع سیگموئید تقریب زد. می‌دانیم که شبکه‌های عصبی ویژگی تقریبی جامع دارند. بنابه این حقیقت، از توانایی شبکه‌های عصبی در تقریب تابع، برای تقریب زدن توابع حالت، کنترل و الحاقی برای مساله کنترل بهینه (۴.۴)-(۱.۴) استفاده می‌کنیم.

حال، طرح تقریبی پیشنهاد شده را برای حل معادلات به دست آمده در (۷.۴)-(۵.۴) و (۱۳.۴)-(۱۱.۴) بکار ببریم. برای هر تابع سه شبکه عصبی در نظر می‌گیریم: حالت (شبکه عصبی آن  $n_x$  است)، کنترل (شبکه عصبی آن  $n_u$  است) و ضرایب لاگرانژ (شبکه‌های عصبی آن  $n_\lambda$  و  $n_\mu$  است) که به صورت زیر داده شده است:

$$\begin{cases} n_x = \sum_{i=1}^I v_x^i \bar{\sigma}(z_x^i), & z_x^i = w_x^i t + b_x^i, \\ n_\lambda = \sum_{i=1}^I v_\lambda^i \bar{\sigma}(z_\lambda^i), & z_\lambda^i = w_\lambda^i t + b_\lambda^i, \\ n_\mu = \sum_{i=1}^I v_\mu^i \bar{\sigma}(z_\mu^i), & z_\mu^i = w_\mu^i t + b_\mu^i, \\ n_u = \sum_{i=1}^I v_u^i \bar{\sigma}(z_u^i), & z_u^i = w_u^i t + b_u^i, \end{cases} \quad (16.4)$$

برای  $i = 1, 2, \dots, I$  که  $I$  تعداد نرون‌هایی است که می‌تواند برای هر شبکه عصبی متفاوت باشد و  $\bar{\sigma}(z) = (t_f - t_0)\sigma(z) + t_0$ .

می‌توانیم بین جواب شرایط بهینگی (۱۲.۴)-(۵.۴) و جواب مساله بهینه سازی نامقید هم ارز آن با جواب‌های آزمایشی ارتباط برقرار کنیم [۱۳، ۵۳]. این توابع آزمایشی به صورت مجموع دو بخش نوشته شده و بر اساس دو حقیقت ساخته می‌شود: قسمت اول شامل پارامترهای قابل قبول است که در شرایط اولیه و مرزی صدق می‌کند. قسمت دوم تک خروجی شبکه عصبی فیدفوروارد با پارامترهای قابل قبول و ورودی واحد شبکه عصبی فیدفوروارد با بردارهای ورودی را بکار می‌گیرد. یادآور می‌شویم که در جمله‌ی دوم، وزن‌ها برای مساله بهینه سازی تنظیم شده‌اند و برای شرایط اولیه و مرزی ساخته نشده‌اند. بر اساس این حقیقت و با استفاده از شبکه عصبی پیشنهاد شده در (۱۶.۴)، جواب‌های آزمایشی (برای

توابع حالت، کنترل و ضرایب لاگرانژ) می‌تواند به صورت زیر تعریف شود:

$$\begin{cases} x_T = x_o + (t - t_o)n_x, \\ \lambda_T = \frac{\partial \Phi}{\partial x_{iT}}|_{t_f} + (t - t_f)n_\lambda, \\ \mu_T = n_\mu, \\ u_T = n_u. \end{cases} \quad (۱۷.۴)$$

جواب‌های آزمایشی (۱۷.۴) یک تقریب جامع است و باید در شرایط (۱۲.۴)-(۵.۴) صدق کند. بنابراین داریم:

$$\dot{x}_{iT} - f_i = 0, \quad i = 1, 2, \dots, n, \quad (۱۸.۴)$$

$$\dot{\lambda}_{iT} + \frac{\partial L_T}{\partial x_{iT}} + \sum_{i=1}^n \frac{\partial f_{iT}}{\partial x_{iT}} \lambda_{iT} + \sum_{l=1}^p \frac{\partial c_{lT}}{\partial x_{iT}} \mu_{lT} = 0, \quad s = 1, 2, \dots, n, \quad (۱۹.۴)$$

$$\frac{\partial L_T}{\partial u_{iT}} + \sum_{i=1}^n \frac{\partial f_{iT}}{\partial u_{sT}} \lambda_{iT} + \sum_{l=1}^p \frac{\partial c_{lT}}{\partial u_{sT}} \mu_{lT} = 0, \quad i = 1, 2, \dots, m, \quad (۲۰.۴)$$

$$\phi_{FB}^{\varepsilon}(\mu_{lT}, c_{lT}) = 0, \quad l = 1, 2, \dots, p, \quad (۲۱.۴)$$

$$x_{iT}(t_o) - x_{oi} = 0, \quad i = 1, 2, \dots, n, \quad (۲۲.۴)$$

$$\lambda_{iT}(t_f) - \frac{\partial \Phi}{\partial x_{iT}}|_{t_f} = 0, \quad i = 1, 2, \dots, n. \quad (۲۳.۴)$$

با دوباره فرموله بندی (۲۳.۴)-(۱۸.۴) به عنوان یک مساله بهینه سازی نامقید، در ابتدا دستگاه بهینه سازی (۲۳.۴)-(۱۸.۴) را با  $\hat{m}$  نقطه  $t_k, k = 1, 2, \dots, q$  در بازه  $[t_o, t_f]$  گسسته سازی می‌کنیم و مساله بهینه سازی را به صورت زیر تعریف می‌کنیم:

$$\text{minimize}_y E(y) = \frac{1}{q} \sum_{k=1}^m \{E_1(t_k, y) + E_2(t_k, y) + E_3(t_k, y) + E_4(t_k, y)\}, \quad (۲۴.۴)$$

که در آن  $y = (w_x, w_\lambda, w_\mu, w_u, b_x, b_\lambda, b_\mu, b_u, v_x, v_\lambda, v_\mu, v_u)^T \in \mathbb{R}^{4I(2n+m+p)}$  و

$$\begin{cases} E_1(t_k, y) = [\dot{x}_{iT} - f_i]^T, & k = 1, 2, \dots, q, \quad i = 1, 2, \dots, n, \\ E_2(t_k, y) = \left[ \dot{\lambda}_{iT} + \frac{\partial L_T}{\partial x_{iT}} + \sum_{i=1}^n \frac{\partial f_{iT}}{\partial x_{iT}} \lambda_{iT} + \sum_{l=1}^p \frac{\partial c_{lT}}{\partial x_{iT}} \mu_{lT} \right]^T, & k = 1, 2, \dots, q, \quad i = 1, 2, \dots, n, \\ E_3(t_k, y) = \left[ \frac{\partial L_T}{\partial u_{sT}} + \sum_{i=1}^n \frac{\partial f_{iT}}{\partial u_{sT}} \lambda_{iT} + \sum_{l=1}^p \frac{\partial c_{lT}}{\partial u_{sT}} \mu_{lT} \right]^T, & k = 1, 2, \dots, q, \quad s = 1, 2, \dots, m, \\ E_4(t_k, y) = [\phi_{FB}^{\varepsilon}(\mu_{lT}, c_{lT})]^T, & k = 1, 2, \dots, q, \quad l = 1, 2, \dots, p. \end{cases} \quad (۲۵.۴)$$

لم ۱.۳.۴. اگر  $y = (w_x^*, w_\lambda^*, w_\mu^*, w_u^*, b_x^*, b_\lambda^*, b_\mu^*, b_u^*, v_x^*, v_\lambda^*, v_\mu^*, v_u^*)^T$  در معادلات زیر صدق کند

$$\eta(y) = \begin{bmatrix} E_1(t_1, y) \\ \vdots \\ E_1(t_m, y) \\ E_2(t_1, y) \\ \vdots \\ E_2(t_m, y) \\ E_3(t_1, y) \\ \vdots \\ E_3(t_m, y) \\ E_4(t_1, y) \\ \vdots \\ E_4(t_m, y) \end{bmatrix} = 0, \quad (26.4)$$

در اینصورت  $y^*$  جواب بهینه (۲۴.۴) است.

برهان. فرض کنیم  $\eta(y^*) = 0$ . در اینصورت برای  $i = 1, 2, 3, 4$  و  $k = 1, 2, \dots, m$ ،  $E_i(\tau_k, y^*) = 0$  چون در (۲۴.۴)،  $E(y) \geq 0$  بنابراین  $y^*$  جواب بهینه (۲۴.۴) است. □

بنابرم (۱.۳.۴) می‌توان بررسی کرد که مساله بهینه سازی (۲۴.۴) با مساله زیر هم ارز است

$$\text{minimize}_y E(y) = \frac{1}{2} \|\eta(y)\|^2 \quad (27.4)$$

برای حل (۲۷.۴)، که یک مساله بهینه سازی نامقید است، می‌توان از الگوریتم‌های بهینه سازی مانند تندترین کاهش، نیوتن، شبه نیوتن، گرادیان مزدوج و غیره استفاده کرد [۲۰، ۶۵، ۳۳]. همچنین می‌توانیم از الگوریتم‌های ابتکاری مانند الگوریتم ژنتیک، بهینه سازی گروه پرندگان، الگوریتم مورچگان و غیره استفاده کنیم [۳۹].

## ۴.۴ مثال‌های عددی

در این بخش، سعی می‌کنیم الگوریتم پیشنهادی را با حل چند مثال نشان دهیم. در همه مثال‌ها ۵ پارامتر برای هر ورودی، خروجی و وزن بایاس استفاده می‌کنیم. در همه مثال‌ها از نرم افزار GAMS برای شبیه‌سازی استفاده کرده‌ایم.

مثال ۱۰.۴.۴. مساله بهینه سازی زیر را در نظر بگیرید:

$$\text{maximize } \int_0^1 u(t) dt,$$

subject to

$$\dot{x}(t) = u(t),$$

$$x(0) = 1, x(1) \text{ is free},$$

$$u(t) \geq 0,$$

$$x(t) - u(t) \geq 0.$$

همیلتونین را به صورت زیر می‌نویسیم

$$H = u + \lambda u + \mu_1 u + \mu_2(x - u),$$

و از شرایط بهینگی (۱۲.۴)-(۵.۴) داریم:

$$\begin{cases} \mu_2 = -\lambda \\ 1 + \lambda + \mu_1 - \mu_2 = 0 \\ \dot{x} = u \\ \mu_1 \geq 0, \mu_1 u = 0 \\ \mu_2 \geq 0, \mu_2(x - u) = 0 \end{cases}$$

چون  $x$  در  $t = 1$  آزاد است داریم  $\lambda(1) = 0$ . حال با حل دستگاه معادله دیفرانسیل فوق، جواب دقیق متغیر حالت و کنترل به صورت زیر است:

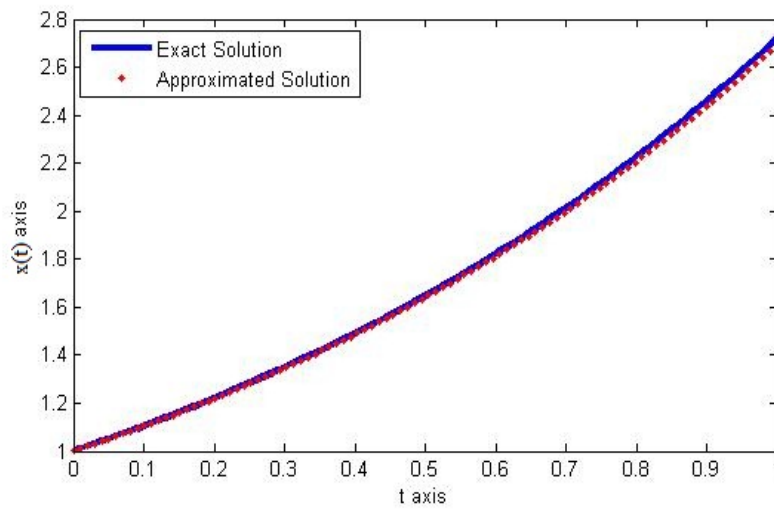
$$\begin{cases} x(t) = \exp(t), \\ u(t) = \exp(t). \end{cases}$$

با در نظر گرفتن شرط اولیه و انتهایی، می‌توانیم جواب‌های آزمایشی را به صورت زیر انتخاب کنیم:

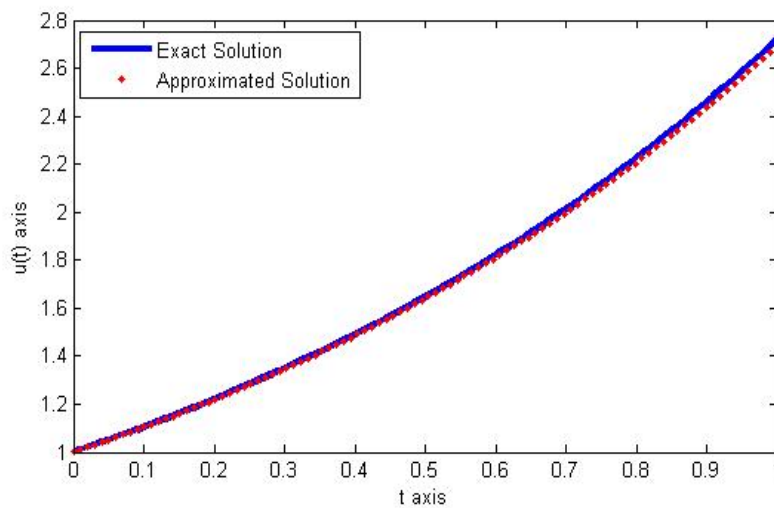
$$\begin{cases} x_T = 1 + tn_x, \\ \lambda_T = (t - 1)n_\lambda, \\ u_T = n_u, \\ \mu_{1T} = n_{\mu_1}, \\ \mu_{2T} = n_{\mu_2}. \end{cases}$$

می‌توانیم جواب‌های تقریبی برای  $x(t)$  و  $u(t)$  را به ترتیب در شکل‌های (۲.۴) و (۳.۴) مشاهده

کنیم.



شکل ۲.۴: تابع حالت برای مثال (۱.۴.۴)



شکل ۳.۴: تابع کنترل برای مثال (۱.۴.۴)

مثال ۲.۴.۴. مساله بهینه سازی زیر را در نظر بگیرید:

$$\text{minimize } -\ln(2) \int_0^1 x(t) dt,$$

subject to

$$\dot{x}(t) = \ln(2)(x(t) - u(t)),$$

$$x(0) = 0, \quad x(1) \text{ is free},$$

$$|u(t)| \leq 1,$$

$$x(t) + u(t) \leq 2.$$



همیلتونین را به صورت زیر می‌نویسیم

$$H = -\ln(2)x + \lambda \ln(2)(x + u) + \mu_1(u - 1) + \mu_2(-u - 1) + \mu_3(x + u - 2)$$

و از شرایط بهینگی (۱۲.۴)-(۵.۴) داریم:

$$\begin{cases} -\ln(2) + \lambda \ln(2) + \mu_3 = -\dot{\lambda} \\ \lambda \ln(2) + \mu_1 - \mu_2 + \mu_3 = 0 \\ \dot{x} = \ln(2)(x + u) \\ \mu_1 \geq 0, \mu_1(u - 1) = 0 \\ \mu_2 \geq 0, \mu_2(-u - 1) = 0 \\ \mu_3 \geq 0, \mu_3(x + u - 2) = 0 \end{cases}$$

چون  $x$  در  $t = 1$  آزاد است داریم  $\lambda(1) = 0$ . حال با حل دستگاه معادله دیفرانسیل فوق، جواب دقیق متغیر حالت و کنترل به صورت زیر است:

$$\begin{cases} x(t) = A \exp((\ln 2)t) - 1, \\ u(t) = 1. \end{cases}$$

و با استفاده از شرایط اولیه و انتهایی داریم  $A = 1$ .

با در نظر گرفتن شرط اولیه و انتهایی، می‌توانیم جواب‌های آزمایشی را به صورت زیر انتخاب کنیم:

$$\begin{cases} x_T = t n_x, \\ \lambda_T = (t - 1) n_\lambda, \\ u_T = n_u, \\ \mu_{1T} = n_{\mu_1}, \\ \mu_{2T} = n_{\mu_2}, \\ \mu_{3T} = n_{\mu_3}. \end{cases}$$

می‌توانیم جواب‌های تقریبی برای  $x(t)$  و  $u(t)$  را به ترتیب در شکل‌های (۴.۴) و (۵.۴) مشاهده کنیم.

مثال ۳.۴.۴. مساله بهینه سازی زیر را در نظر بگیرید:

$$\text{minimize } \int_0^3 x_1(t) dt,$$

subject to

$$\dot{x}_1(t) = x_2(t),$$

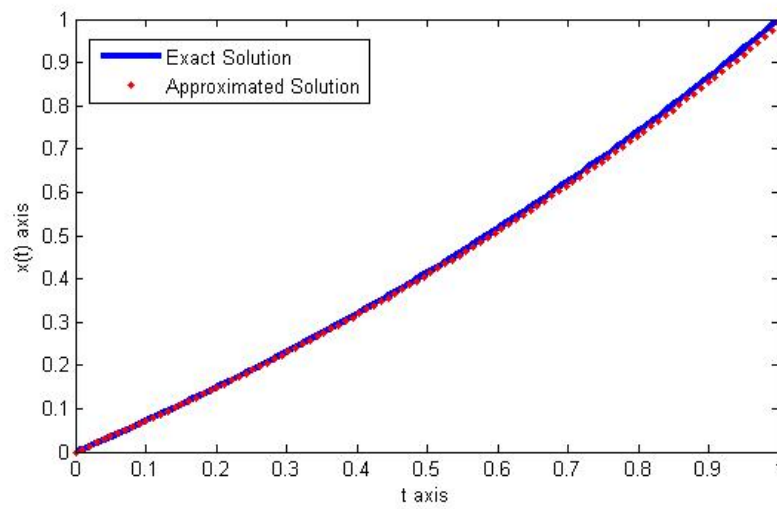
$$\dot{x}_2(t) = u(t),$$

$$x_1(0) = 2, x_1(3) = \text{is free},$$

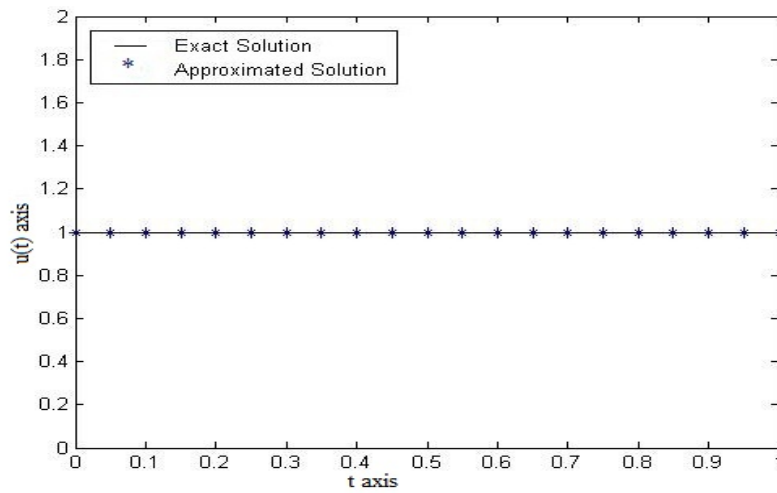
$$x_2(0) = 0, x_2(3) = \text{is free},$$

$$|u(t)| \leq 2,$$

$$x_1 \geq -7.$$



شکل ۴.۴: تابع حالت برای مثال (۲.۴.۴)



شکل ۵.۴: تابع کنترل برای مثال (۲.۴.۴)

هامیلتونین را به صورت زیر می‌نویسیم

$$H = \gamma x_1 + \lambda_1 x_2 + \lambda_2 u + \mu_1 (u - \gamma) + \mu_2 (-u - \gamma) + \mu_3 (-\gamma - x_1)$$

و از شرایط بهینگی (۱۲.۴) - (۵.۴) داریم:

$$\begin{cases} 2 - \mu_3 = -\lambda_1 \\ \lambda_1 = -\lambda_2 \\ \lambda_2 + \mu_1 - \mu_2 = 0 \\ \dot{x}_1 = x_2 \\ \dot{x}_2 = u \\ \mu_1 \geq 0, \quad \mu_1(u - 2) = 0 \\ \mu_2 \geq 0, \quad \mu_2(-u - 2) = 0 \\ \mu_3 \geq 0, \quad \mu_3(-7 - x_1) = 0 \end{cases}$$

چون  $x_1, x_2$  در  $t = 3$  آزاد است داریم  $\lambda_1(3) = \lambda_2(3) = 0$ . حال با حل دستگاه معادله دیفرانسیل فوق، جواب دقیق متغیر حالت و کنترل به صورت زیر است:

$$\begin{cases} x_1(t) = A - t^2, \\ x_2(t) = B - 2t, \\ u(t) = -2. \end{cases}$$

و با استفاده از شرایط اولیه و انتهایی داریم  $A = 2, B = 0$ . با در نظر گرفتن شرط انتهایی آزاد و شرط اولیه  $x_1(0) = 2, x_2(0) = 0$  می‌توانیم جواب‌های آزمایشی را به صورت زیر انتخاب کنیم:

$$\begin{cases} x_{1T} = 2 + tn_{x_1}, \\ x_{2T} = tn_{x_2}, \\ \lambda_{1T} = (t - 3)n_{\lambda_1}, \\ \lambda_{2T} = (t - 3)n_{\lambda_2}, \\ u_T = n_u, \\ \mu_{1T} = n_{\mu_1}, \\ \mu_{2T} = n_{\mu_2}, \\ \mu_{3T} = n_{\mu_3}. \end{cases}$$

می‌توانیم جواب‌های تقریبی برای  $x_1(t), x_2(t)$  و  $u(t)$  را به ترتیب در شکل‌های (۶.۴)، (۷.۴) و (۸.۴) مشاهده کنیم.

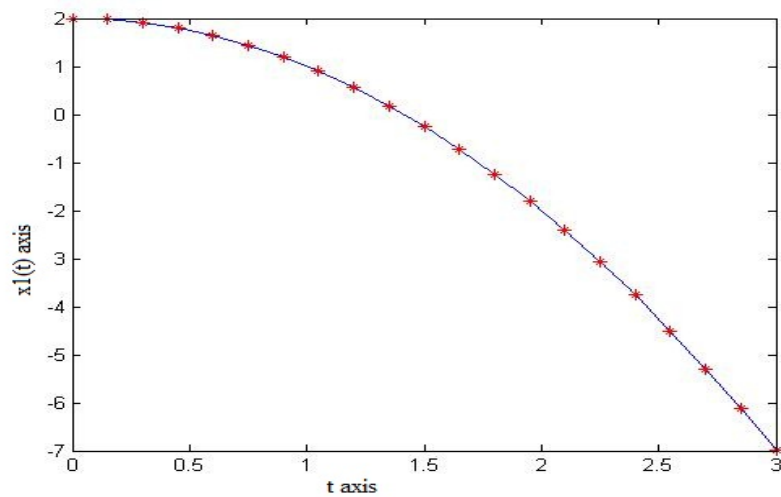
مثال ۴.۴.۴. شکل کلی طراحی مسیر بهینه برای یک جسم صلب و هدف پروازی آزاد  $\mathcal{A}$ ، که به عنوان یک نقطه در نظر گرفته می‌شود، می‌تواند به صورت مساله بهینه زیر در نظر گرفته شود:

$$\text{minimize } I(q(t), \dot{q}(t)) = \int_0^T F(t, q(t), \dot{q}(t)) dt \quad (28.4)$$

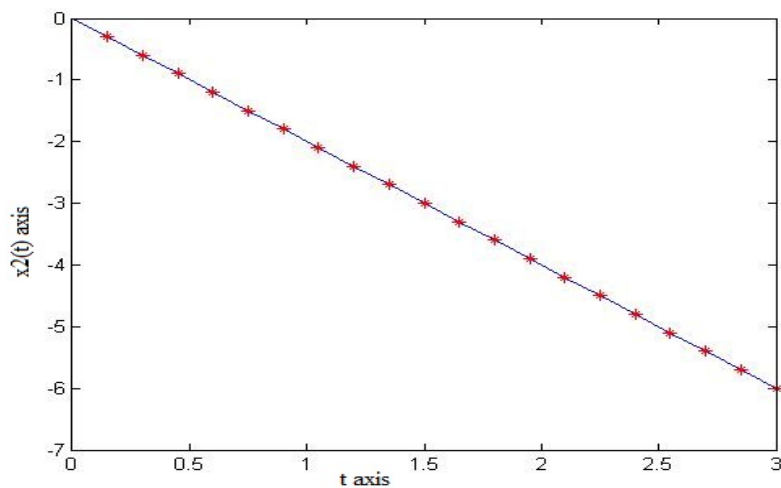
subject to

$$\varphi_i(t, q(t), \dot{q}(t)) > 0, \quad i = 1, 2, \dots, s, \quad (29.4)$$

$$q(0) = q_0, \quad q(T) = q_T, \quad (30.4)$$



شکل ۶.۴: تابع حالت برای مثال (۳.۴.۴)



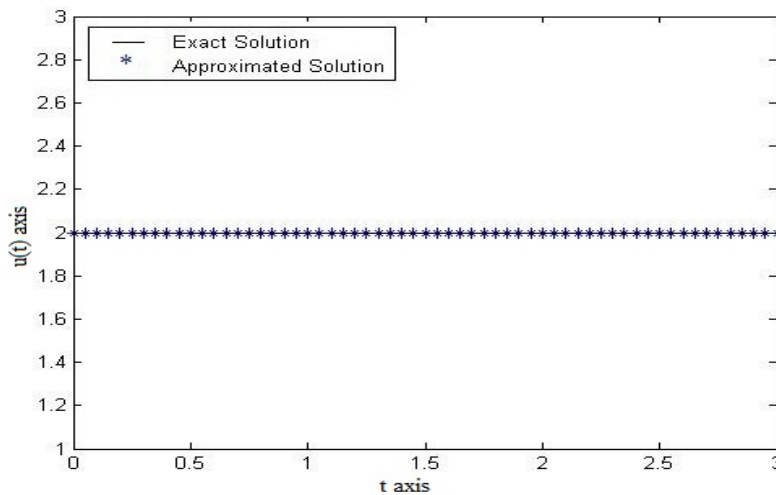
شکل ۷.۴: تابع حالت برای مثال (۳.۴.۴)

که  $q$  یک شکل از فضای زمانی  $\mathcal{C}$  است، که آن را به عنوان موقعیت هدف  $\mathcal{A}$  در نظر می‌گیریم،  $I$  تابعی است که می‌تواند برخی ویژگی‌های هندسی-حرکتی به عنوان طول مسیر را نشان می‌دهد و  $i$  امین نامساوی در (۲۹.۴) با تصویر مکمل  $i$  امین مانع  $\mathcal{B}_i(t)$  در  $\mathcal{C}$  هم ارز است که مکمل

$$\mathcal{C} \mathcal{B}_i(t) = \{q : \mathcal{A}(q) \cap \mathcal{B}_i(t) \neq \emptyset\} \quad \forall t \in [0, T]$$

است.

در تمام مثال‌ها فرض می‌کنیم که موانع به عنوان دایره‌ها با شعاع  $r_i$ ،  $i = 1, 2, \dots, s$  در صفحه در نظر گرفته می‌شوند. اگر فرض کنیم  $i$  امین مانع در درگشتاور  $\tau$  دارای شکل هندسی دایره نباشد، مانند  $\gamma_i = \mathcal{B}_i(\tau)$  با مرز فشرده  $\partial\gamma_i$ ، در اینصورت یک پوشش  $\partial\gamma_i$  وجود دارد که بوسیله تعداد متناهی

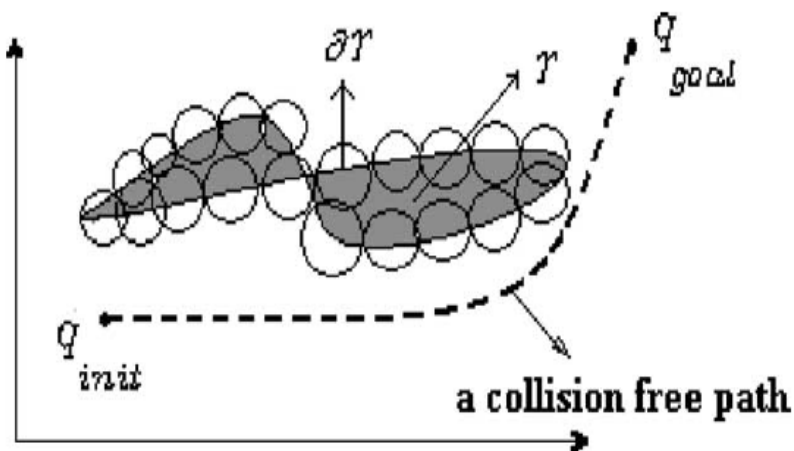


شکل ۸.۴: تابع کنترل برای مثال (۳.۴.۴)

دایره مانند  $C_l, l = 1, 2, \dots, L$ ، بطوریکه  $C_l \subseteq \mathcal{C} \mathcal{B}_i(\tau)$  پوشیده می‌شود (شکل (۹.۴) را نگاه کنید). بنابراین می‌توانیم تعدادی شرط در مساله بهینه سازی بجای موانع  $\mathcal{B}_i(\tau)$  جایگزین کنیم. حال کنترل مصنوعی  $u(\cdot)$  را به صورت زیر تعریف می‌کنیم:

$$\dot{q}(t) = u(t),$$

مساله کنترل بهینه آمیخته زیر، مساله کنترل بهینه (۳۰.۴)–(۲۸.۴) را نشان می‌دهد.



شکل ۹.۴: مانع  $\gamma$  و مرز آن که توسط دایره‌ها پوشیده شده است

$$\text{minimize } I(q(t), u(t)) = \int_0^T F(t, q(t), u(t)) dt \quad (31.4)$$

subject to

$$\dot{q}(t) = u(t), \quad (32.4)$$

$$\varphi_i(t, q(t), u(t)) > 0, \quad i = 1, 2, \dots, q, \quad (33.4)$$

$$q(0) = q_0, \quad q(T) = q_T. \quad (34.4)$$

که در آن  $q(t), u(t) \in \mathbb{R}^n$  و  $t \in [0, T]$ .

به عنوان نمونه، در این مثال برنامه ریزی مسیر بهینه با موانع متحرک را در نظر بگیرید. مساله کنترل بهینه متناظر را به مساله اصلی به صورت زیر تبدیل می‌کنیم:

$$\text{minimize } \int_0^1 \sqrt{(u_1(t))^2 + (u_2(t))^2} dt,$$

subject to

$$\dot{q}_1(t) = u_1(t),$$

$$\dot{q}_2(t) = u_2(t),$$

$$\varphi_i(t, q(t), u(t)) > 0, \quad i = 1, 2, \dots, q$$

$$q_1(0) = 0, \quad q_2(0) = 1$$

و

$$\varphi_i(t, q(t), u(t)) = (q_1(t) - \alpha_i)^2 + (q_2(t) - \beta_i)^2 - r_i^2$$

مساله را به دو صورت زیر در نظر می‌گیریم:

حالت اول: در این حالت یک مانع متحرک به صورت زیر در نظر می‌گیریم:

$$\varphi(t, q(t), u(t)) = (q_1(t) - \alpha)^2 + (q_2(t) - \beta)^2 - r^2$$

که در آن

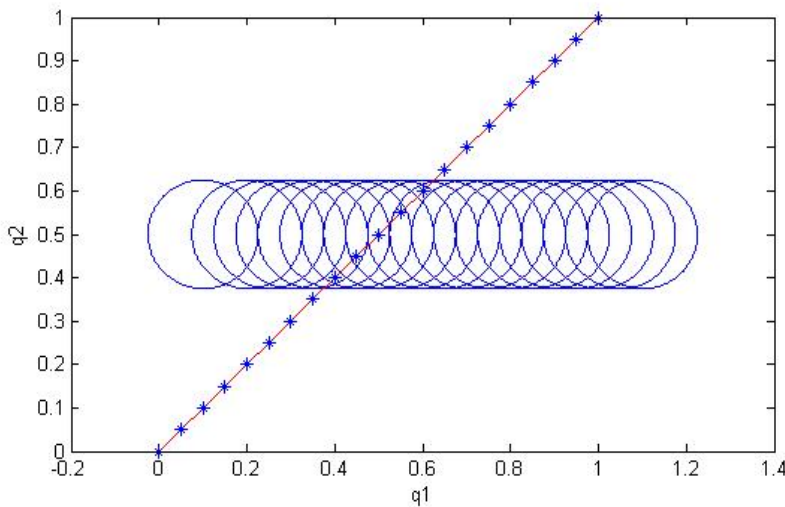
$$\begin{cases} \alpha(t) = t, \\ \beta(t) = 0.5, \\ r = \frac{1}{\lambda}. \end{cases}$$

حالت دوم: در این حالت دو مانع متحرک به صورت زیر در نظر می‌گیریم:

$$\varphi_i(t, q(t), u(t)) = (q_1(t) - \alpha_i)^2 + (q_2(t) - \beta_i)^2 - r_i^2, \quad i = 1, 2$$

که در آن

$$\begin{cases} \alpha_1(t) = t, \\ \beta_1(t) = 0.5, \\ r_1 = \frac{1}{\lambda}. \end{cases}, \quad \begin{cases} \alpha_1(t) = t, \\ \beta_1(t) = -1/2t^2 + 1/2t + 0.7, \\ r_1 = \frac{1}{\lambda}. \end{cases}$$



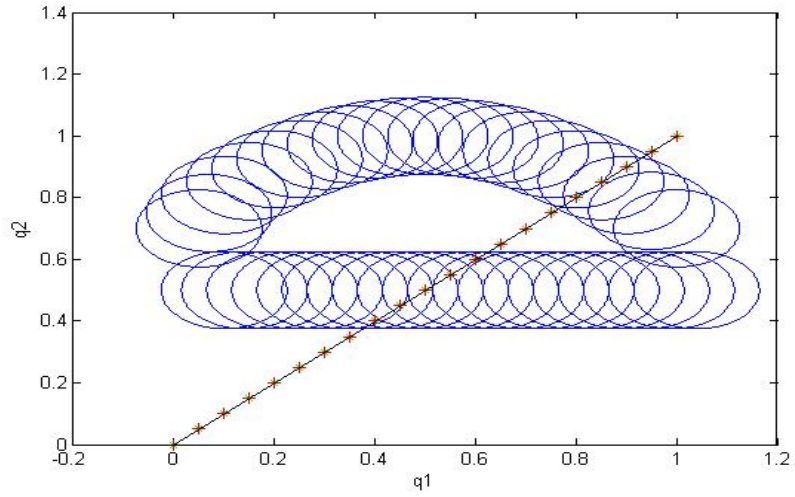
شکل ۱۰.۴: مسیر بهینه تقریبی برای حالت اول برای مثال (۴.۴.۴)

با در نظر گرفتن شرایط اولیه و انتهایی، می‌توانیم جواب‌های آزمایشی را به صورت زیر انتخاب کنیم:

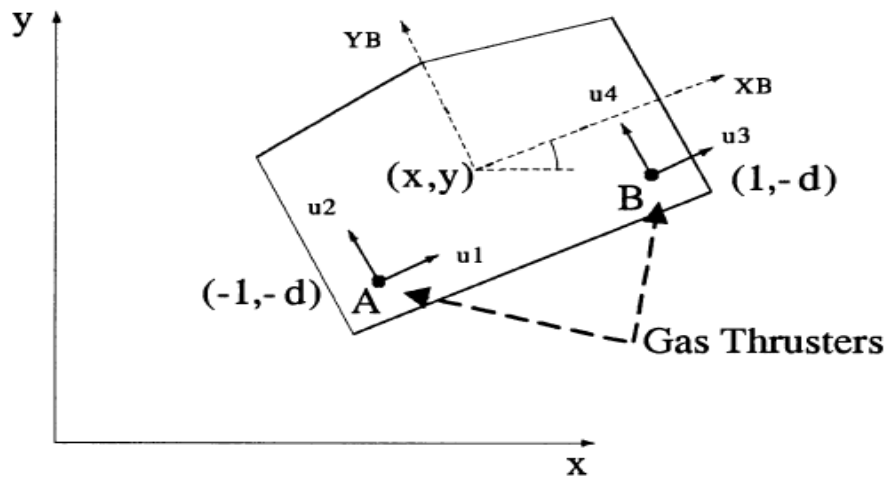
$$\begin{cases} q_{1T} = t + t(t-1)n_{q_1}, \\ q_{2T} = t + t(t-1)n_{q_2}, \\ \lambda_{1T} = n_{\lambda_1}, \\ \lambda_{2T} = n_{\lambda_2}, \\ u_{1T} = n_{u_1}, \\ u_{2T} = n_{u_2}. \end{cases}$$

مسیر بهینه تقریبی و موانع متحرک در حالت اول و دوم به ترتیب در شکل‌های (۱۰.۴) و (۱۱.۴) نشان داده شده است.

مثال ۵.۴.۴. یک ربات شناور مسطح را در نظر بگیرید، که به عنوان یک جسم صلب شناور آزاد روی یک سطح صاف مدل شده و در شکل (۱۲.۴) نشان داده شده است. این ربات می‌تواند با استفاده از نیروی پیشرانه گاز مانور دهد. ربات شناور بوسیله‌ی ۳ مختصات توصیف می‌شود.  $(x, y)$  مختصات دکارتی مرکز آن از جرم نسبت به مختصات اینرسی و  $\theta$  جهت زاویه‌ای مختصات ربات نسبت به محور  $X$  است. دو نیروی پیشرانه گاز که می‌تواند در جت در طول مختصات مثبت  $X_B$  و منفی  $Y_B$  آتش بگیرد، روی ربات شناور در مختصات  $(-l, -d)$  و  $(l, d)$  نصب شده و که در شکل (۱۲.۴) نشان داده شده است. معادلات حرکت ربات به صورت زیر آمده است:



شکل ۱۱.۴: مسیر بهینه تقریبی برای حالت دوم برای مثال (۴.۴.۴)



شکل ۱۲.۴: بدنه‌ی صلب ریات مسطح شناور آزاد



$$\text{minimize } \int_0^{t_f} ((u_1(t))^2 + (u_2(t))^2 + (u_3(t))^2 + (u_4(t))^2) dt,$$

subject to

$$\dot{x}_1(t) = x_2(t),$$

$$\dot{x}_2(t) = \frac{u_1(t) + u_3(t)}{10} \cos(\theta) - \frac{u_2(t) + u_4(t)}{10} \sin(\theta),$$

$$\dot{x}_3(t) = x_4(t),$$

$$\dot{x}_4(t) = \frac{u_1(t) + u_3(t)}{10} \sin(\theta) - \frac{u_2(t) + u_4(t)}{10} \cos(\theta),$$

$$\dot{x}_5(t) = x_6(t),$$

$$\dot{x}_6(t) = \frac{5}{12}(u_1(t) + u_3(t)) + \frac{5}{12}(u_4(t) - u_2(t)),$$

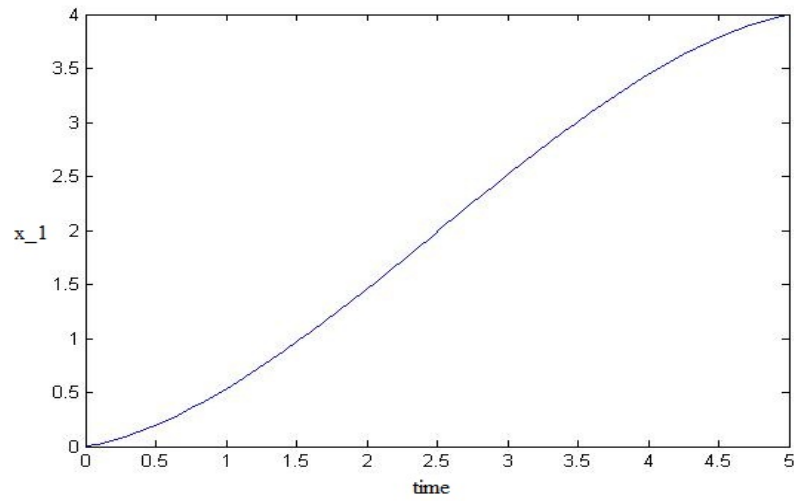
$$x_1(0) = x_2(0) = \dots = x_6(0) = 0,$$

$$x_1(5) = 4, \quad x_2(5) = 0, \quad x_3(5) = 4, \quad x_4(5) = 0, \quad x_5(5) = \frac{\pi}{4}, \quad x_6(5) = 0.$$

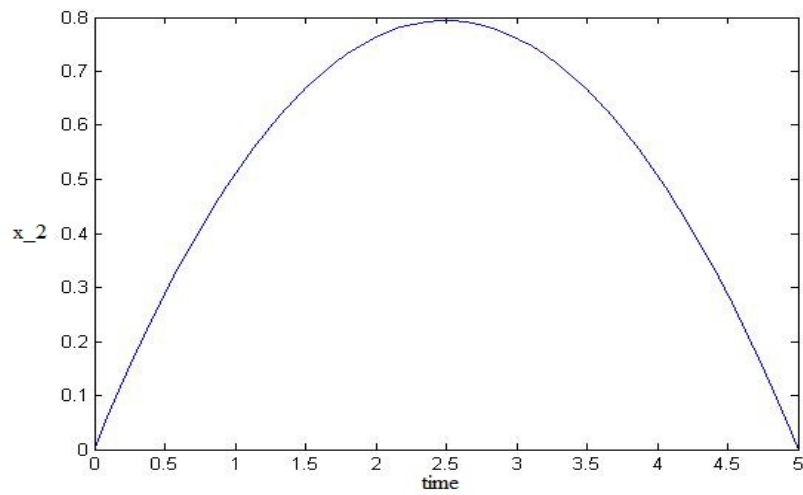
با در نظر گرفتن شرایط اولیه و انتهایی، می‌توانیم جواب‌های آزمایشی را به صورت زیر انتخاب کنیم:

$$\left\{ \begin{array}{l} x_{1T} = 0.8t + t(t-5)n_{x_1}, \\ x_{2T} = t(t-5)n_{x_2}, \\ x_{3T} = 0.8t + t(t-5)n_{x_3}, \\ x_{4T} = t(t-5)n_{x_4}, \\ x_{5T} = 0.2\frac{\pi}{4}t(t-5)n_{x_5}, \\ x_{6T} = t(t-5)n_{x_6}, \\ \lambda_{1T} = n_{\lambda_1}, \\ \lambda_{2T} = n_{\lambda_2}, \\ \lambda_{3T} = n_{\lambda_3}, \\ \lambda_{4T} = n_{\lambda_4}, \\ \lambda_{5T} = n_{\lambda_5}, \\ \lambda_{6T} = n_{\lambda_6}, \\ u_{1T} = n_{u_1}, \\ u_{2T} = n_{u_2}, \\ u_{3T} = n_{u_3}, \\ u_{4T} = n_{u_4}. \end{array} \right.$$

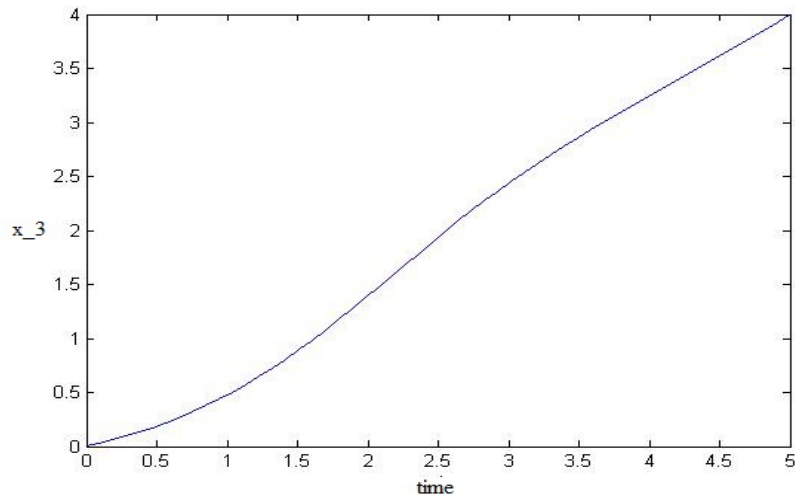
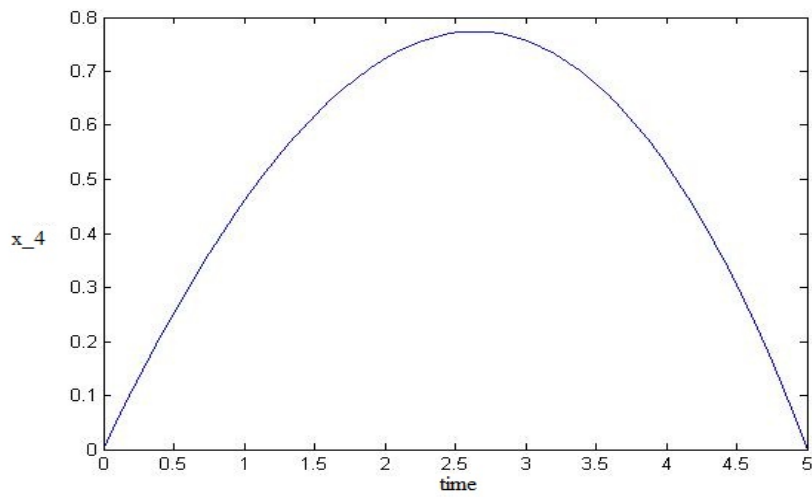
مسیرهای بهینه برای این مثال را میتوان به ترتیب در شکل‌های (۱۳.۴) تا (۱۸.۴) مشاهده کرد.

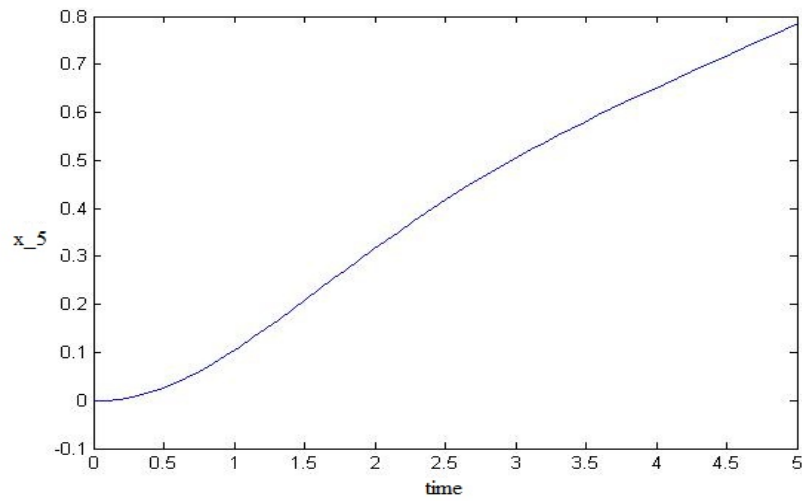


شکل ۱۳.۴: مسیر بهینه  $x_1$  برای مثال (۵.۴.۴)

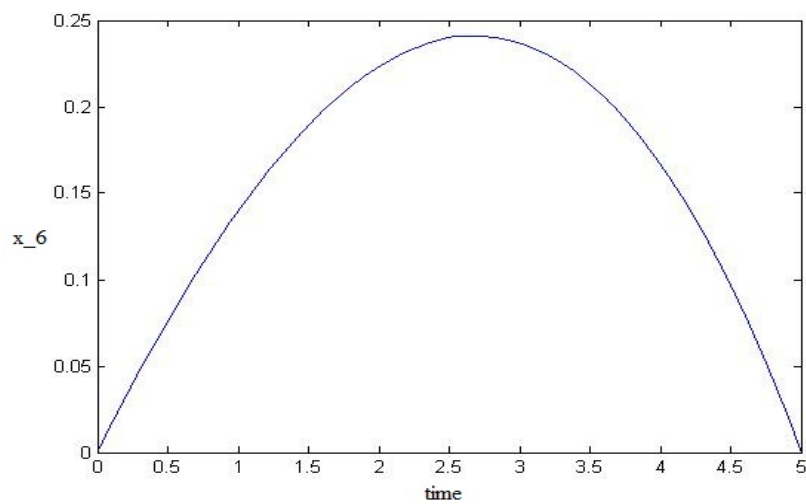


شکل ۱۴.۴: مسیر بهینه  $x_2$  برای مثال (۵.۴.۴)

شکل ۱۵.۴: مسیر بهینه  $x_3$  برای مثال (۵.۴.۴)شکل ۱۶.۴: مسیر بهینه  $x_4$  برای مثال (۵.۴.۴)



شکل ۱۷.۴: مسیر بهینه  $x_5$  برای مثال (۵.۴.۴)



شکل ۱۸.۴: مسیر بهینه  $x_6$  برای مثال (۵.۴.۴)

# نتیجه‌گیری و پیشنهادات

## نتیجه‌گیری

با توجه به دشواری‌های حل مسائل کنترل بهینه به روش تشریحی، مسائل کنترل بهینه را می‌توان با استفاده از یک مدل تقریبی شبکه عصبی حل کرد. در این پایان نامه مسائل کنترل بهینه و مسائل کنترل بهینه با قیدهای نامساوی را معرفی و شرایط بهینگی برای این نوع مسائل را بیان کرده و به حل آن به روش تشریحی و روش شبکه‌های عصبی پرداختیم. نتایج شبیه سازی شده نشان از کارایی شبکه‌های عصبی در حل مسائل کنترل بهینه دارد.

## پیشنهادات

- استفاده از شبکه‌های عصبی فیدفوروارد در حل مسائل کنترل بهینه تاخیری.
- استفاده از شبکه‌های عصبی فیدفوروارد در حل مسائل افق نامتناهی.
- استفاده از شبکه‌های عصبی فیدفوروارد در حل مسائل مقدار اولیه و مقدار مرزی.
- استفاده از شبکه‌های عصبی فیدفوروارد در حل مسائل ناهموار.



# مراجع

- [۱] ضمیری، آذر. شبکه‌های عصبی مصنوعی و بهینه‌سازی. پایان‌نامه کارشناسی ارشد، دانشکده علوم ریاضی، دانشگاه فردوسی مشهد، آبان ۱۳۸۷.
- [۲] مقدس، محمد، عفتی، سهراب، و اسحاق‌نژاد، محمد. یک شبکه عصبی برای حل مسائل بهینه‌سازی غیرخطی نامحدب. در ششمین کنفرانس بین‌المللی
- [۳] شبکه‌های عصبی، فیلیپ پیکتن، ترجمه دکتر غضنفری
- [۴] منطق فازی و شبکه‌های عصبی، اس.وی. کارتالاپوس، ترجمه دکتر جورابیان
- [۵] محاسبات نرم، رضا حسینقلی زاده
- [۶] آشنایی با شبکه‌های عصبی، دکتر محمود البرزی
- [۷] مبانی شبکه‌های عصبی، تالیف دکتر منہاج
- [۸] آ. کرایه‌چیان، معادلات دیفرانسیل و کاربرد آنها، ۱۳۸۴.
- [۹] مهرجو، احمد، شبکه‌های عصبی مصنوعی، دانشگاه آزاد اسلامی واحد اسلامشهر، اسلامشهر، ۱۳۸۶.
- [10] A. Bouzerdoum and T. R. Pattison. Neural network for quadratic optimization with bound constraints. *IEEE Trans. Neural Netw.*, 4(2):293–304, 1993.
- [11] A. Cichocki, R. Unbehauen, K. Weizierl and R. Holzl, A new neural network for solving linear programming problems, *European Journal of Operational Research* 93, 244–245, 1996
- [12] A. Lapedes, R. Farber, How neural nets work? in: D.Z. Anderson (Ed.), *Neural Information Processing Systems*, AIP, pp. 442–456, 1988.
- [13] A. Malek, R. Shekari Beidokhti, Numerical solution for high order differential equations using a hybrid neural network Optimization method *Applied Mathematics and Computation*, 183 :260–271, 2006.
- [14] A. Rodriguez-Vazquez, R. Dominguez-Castro, A. Rueda, J.L. Huertas and E. Sanchez Sinencio. Nonlinear switched capacitor ‘neural’ networks for optimization problems. *IEEE Trans. Circuits Syst.*, 37(3):384–398, 1990.

- [15] Bo-An Liu, B. Jammes, Solving ordinary differential equations by neural networks, in: Proceeding of 13th European Simulation Multi-Conference Modelling and Simulation: A Tool for the Next Millennium, Warsaw, Poland, June 1–4, 1999.
- [16] B. Widrow. Generalization and information storage in networks of adaline ‘neurons’. in *Self Organizing Systems*, pp. 435–461. Spartan Books, 1959.
- [17] B. Widrow and M. E. Hoff. Adapting switching circuits. in 1960 ire wescon Convention Record, no. 4, pp. 96–104, New York, NY, 1960. ire.
- [18] B. Widrow and M. A. Lehr. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proc. IEEE*, 78(9):1415–1442, 1990.
- [19] B. Widrow and R. Winter. Neural nets for adaptive filtering and adaptive pattern recognition. *Computer*, 21(3):25–39, 1988.
- [20] Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (2006). *Nonlinear Programming-Theory and Algorithms*. Third edition, John Wiley & Sons, Hoboken, New Jersey.
- [21] C.Y. Maa and M. A. Schanblatt, A two-phase optimization neural network. *IEEE Trans. Neural Netw.*, 3(6):1003–1009, 1992.
- [22] D. Tank and J. J. Hopfield. Simple ‘neural’ optimization networks: An a/d converter, signal decision circuit, and a linear programming circuit. *IEEE Trans. Circuits Syst.*, 33(5):533–541, 1986.
- [23] D. O. Hebb. *The Organization of Behavior: A Neuropsychological Theory*. John Wiley & Sons, Inc., New York, 1949.
- [24] D. E. Rumelhart, G. E. Hinton and R. J. Williams. Learning representations by backpropagating errors. *Nature*, 323:533–536, 1986.
- [25] D.E. Kirk, *Optimal Control Theory: An Introduction*, Englewood Cliffs, New York, 1970.
- [26] D. Sun, and J. Sun. Strong semismoothness of the Fischer–Burmeister SDC and SOC complementarity functions, *Source, Mathematical Programming* 103 (3): 575–581, 2005.
- [27] F. Rosenblatt. *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. Report (Cornell Aeronautical Laboratory). Spartan Books, 1962.
- [28] F. Rosenblatt. *Neurocomputing: foundations of research*. chap. The perception: a probabilistic model for information storage and organization in the brain, pp. 89–114. MIT Press, Cambridge, MA, USA, 1988.



- [29] F. Facchinei, H. Jiang, and Qi L. A smoothing method for mathematical programs with equilibrium constraints. *Mathematical Programming* 35: 107- 134, 1999.
- [30] G. Cybenko, Approximation by superpositions of a sigmoidal function, *Math. Control Signals, Syst.* 2, 303–314, 1989.
- [31] Handbook of neural network signal processing, Yu Hen Hu, Jenq-Neng Hwang
- [32] J. A. Freeman ,D. M. Skapura. *Neural Networks (Algorithms, Applications, and Programming Techniques)*
- [33] J. Nocedal, S. Wright, *Numerical Optimization (second ed.)* Berlin, NewYork: Springer-Verlag, 2006.
- [34] J. Stanley, *Introduction to Neural Networks*, third ed., Sierra Mardre, 1990.
- [35] J.J. Mckeown, F. Stella, G. Hall, Some numerical aspects of the training problem for feed-forward neural nets, *Neural Networks* 10 (8), 1455–1463, 1997.
- [36] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci.*, 79(8):2554–2558, 1982.
- [37] J. J. Hopfield and D. W. Tank. “neural” computation of decisions in optimization problems. *Biological Cybernetics*, 52(3):141–152, 1985.
- [38] R. E. Kalman. “Contributions to the Theory of Optimal Control,” *Bof. SOC. Mat. Mex.* 102-119, 1960.
- [39] K. Y. Lee, M. A. El-Sharkawi, *Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems*.
- [40] V. Keckman, *Learning and soft computing*. MIT press, Cambridge, MA, 2001.
- [41] K. Hornick, M. Stinchcombe, H. white, Multilayer feedforward networks are universal approximators, *Neural Networks* 2 (5), 359–366, 1989.
- [42] K. Funahashi, On the approximate realization of continuous mappings by neural networks, *Neural Networks*, 2, 183–192, 1989.
- [43] M. Minsky, S. Papert, *Perceptrons*, MIT Press, 1969.
- [44] M. P. Kennedy and L. O. Chua. Neural networks for nonlinear programming. *IEEE Trans. Circuits Syst.*, 35(5):554–562, 1988.
- [45] M. L. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. Mit Press, 1972.

- [46] M. Moghaddas and S. Effati. A novel recurrent neural network based on ncp function for solving convex nonlinear optimization problems. in The 5 th Iranian Conference on Applied Mathematics, Hamadan, Iran, September 2013. Bu-Ali Sina University.
- [47] M. Moghaddas and M. Eshaghnejad. A recurrent neural network for solving non-convex non linear optimization problem. in The 44 th Annual Iranian Mathematics Conference, Mashhad, Iran, August 2013. Ferdowsi University of Mashhad.
- [48] P. Werbos. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph.D. thesis, Harvard University, Cambridge, MA, 1974.
- [49] P. Picton, Neural Networks, second ed., Palgrave, Great Britain, 2000.
- [50] P. Lawrence, Differential Equations and Dynamical Systems, Springer-Verlag, 2001.
- [51] R.J. Schalkoff, Artificial Neural Networks, McGraw-Hill, New York, 1997.
- [52] R.P. Lippmann, An introduction to computing with neural nets, IEEE ASSP Magazine, 4–22, 1987.
- [53] R. Shekari Beidokhti, A. Malek, Solving initial-boundaryvalueproblemsforsystems of partial differential equations using neural networks and optimization techniques, Journal of the Franklin Institute, 346: 898- 913, 2009.
- [54] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, New Jersey, 1999.
- [55] S. Effati and M. A. Baymani. new nonlinear neural network for solving convex nonlinear programming problems. Appl. Math. and Comput., 168(2):1370 – 1379, 2005.
- [56] S. Effati, A. Ghomashi and M. Abbasi. A novel recurrent neural network for solving mlcps and its application to linear and quadratic programming. Asia-Pacific Journal of Operational Research, 28(4):523–541, November 2011.
- [57] S. Effati, A. Ghomashi, and A. R. Nazemi. Application of projection neural network in solving convex programming problems. Appl. Math. Comput., 188(2):1103 – 1114, 2007.
- [58] S. Effati, M. Pakdaman. Optimal control problem via neural network. Neural Comput & Applic, 23: 2093-2100, 2013.
- [59] S. K. Agrawal, B. C. Fabien, Optimizationof Dynamic System.
- [60] S. Zhang and A.G. Constantinides. Lagrange programming neural networks. IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., 39(7):441–452, 1992.

- 
- [61] S.-H. Pan, and J.-S. Chen. A semismooth Newton method for the SOCCP based on a one-parametric class of SOC complementarity functions, *Computational Optimization and Applications* 45: 59 -88, 2010.
- [62] S. Haykin, *Neural Networks*.
- [63] T. Khanna, *Foundations of Neural Networks*, Addison-Wesley, Reading, MA, 1990.
- [64] W. McCulloch, and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.*, 5(4):115–133, 1943.
- [65] X-S. Zhang, *Neural Networks in Optimization*, Kluwer Academic Publishers, Dordrecht, Boston, London, 2000.
- [66] Y. Xia, G. Feng and J. Wang. A novel recurrent neural network for solving nonlinear optimization problems with inequality constraints. *IEEE Trans. Neural Netw.*, 19(8):1340–1353, 2008.
- [67] Y. Xia and J.Wang. A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints. *IEEE Trans. Circuits Syst. I, Reg. Papers*, 51(7):1385–1394, 2004.
- [68] Y. Xia, and J. Wang. A recurrent neural network for solving nonlinear convex programs subject to linear constraints. *IEEE Trans. Neural Netw.*, 16(2):379–386, 2005.
- [69] Y. Xia, H. Leung and J. Wang. A projection neural network and its application to constrained optimization problems. *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl*, 49(4):447–458, 2002.



# واژه‌نامه فارسی به انگلیسی

Pontryagin Minimum Principle	اصل مینیمم پونتریاگین
Optimization Algorithm	الگوریتم‌های بهینه سازی
Genetic Algorithm	الگوریتم ژنتیک
Steepest Descent Algorithm	الگوریتم تندترین کاهش
Inertially	اینرسی
Optimization	بهینه سازی
Optimal Path Planning	برنامه ریزی مسیر بهینه
Functional	تابعی
Trajectory Function	تابع مسیر
Control Function	تابع کنترل
Error Function	تابع خطا
Trial Solution	جواب‌های آزمایشی
Optimal Solution	جواب بهینه
Mass Relative	جرم نسبی
Angular Orientation	جهت زاویه‌ای
Variational Calculus	حساب تغییرات
Neuron System	دستگاه عصبی
Newton Methods	روش نیوتن
Quasi Newton Methods	روش شبه نیوتن
Robot	ربات
Neural Network	شبکه عصبی
Artificial Neural Network	شبکه عصبی مصنوعی
Biological Neural Network	شبکه عصبی بیولوژیکی
Optimality Condition	شرایط بهینگی
Initial and Boundary Condition	شرایط اولیه و مرزی

Lagrangian Multipliers	ضرایب لاگرانژ
Optimal Control	کنترل بهینه
Ordinary Differential Equation	معادله دیفرانسیل معمولی
State Variable	متغیر وضعیت (حالت)
Control Variable	متغیر کنترل
Constraint	محدودیت
Unconstraint Optimization Problem	مساله بهینه سازی نامقید
Nonlinear Optimal Control Problem	مساله کنترل بهینه غیر خطی
Obstacle Moving	موانع متحرک
Cordinate	مختصات
Thrusters Gass	نیروی پیشرانه‌ی گاز

# واژه‌نامه انگلیسی به فارسی

Angular Orientation	جهت زاویه‌ای
Artificial Neural Network	شبکه عصبی مصنوعی
Biological Neural Network	شبکه عصبی بیولوژیکی
Control Function	تابع کنترل
Constraint	محدودیت
Control Variable	متغیر کنترل
Cordinate	مختصات
Errore Function	تابع خطا
Functional	تابعی
Genetic Algorithm	الگوریتم ژنتیک
Inertially	اینرسی
Initial and Boundary Condition	شرایط اولیه و مرزی
Lagrangian Multipliers	ضرایب لاگرانژ
Mass Relative	جرم نسبی
Neuron System	دستگاه عصبی
Newton Methods	روش نیوتن
Neural Network	شبکه عصبی
Nonlinear Optimal Control Problem	مساله کنترل بهینه غیر خطی
Optimization Algorithm	الگوریتم‌های بهینه سازی
Optimization	بهینه سازی
Optimal Path Planning	برنامه ریزی مسیر بهینه
Optimal Solution	جواب بهینه
Optimality Conddition	شرایط بهینگی
Optimal Control	کنترل بهینه
Ordinary Differential Equation	معادله دیفرانسیل معمولی

Obstacle Moving	موانع متحرک
Pontryagin Minimum Principle	اصل مینیمم پونترباگین
Quasi Newton Methods	روش شبه نیوتن
Robot	ربات
Steepest Descent Algorithm	الگوریتم تندترین کاهش
State Variable	متغیر وضعیت (حالت)
Trajectory Function	تابع مسیر
Trial Solution	جواب‌های آزمایشی
Thrusters Gass	نیروی پیشرانده‌ی گاز
Unconstraint Optimization Problem	مساله بهینه سازی نامقید
Variational Calculus	حساب تغییرات



## **Aabstract**

In this thesis a class of nonlinear optimal control problem with equality and inequality constrain is considered, and propose a new method based on capabilities of artificial neural networks, in function approximation, to attain the solution of optimal control problems. To do so, we try to approximate the solution of Hamiltonian conditions based on the Pontryagin minimum principle (PMP), then by constructing an error function, that contains all PMP conditions, we define an unconstrained minimization problem. In the proposed error function, we used trial solutions for the trajectory function, control function and the Lagrange multipliers. These trial solutions are constructed by using neurons. Then, we minimize the error function that contains just the weights of the trial solutions. Substituting the optimal values of the weights in the trial solutions, we obtain the optimal trajectory function, optimal control function and the optimal Lagrange multipliers.

**keywords:** Optimal control; Pontryagin minimum principle; Neural networks; Optimization.



**University of Shahrood**

**Faculty Of Mathematical Sciences**

**One application of neural networks for  
solving optimal control problems**

**Rezvan Karami**

**Supervisor**

**Dr Alireza Nazemi**

**Sebtember 2015**