

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه صنعتی شاهرود

دانشکده: ریاضی

گروه: کاربردی

تخصیص مقدار ویژه برای سیستم های ناوردای زمانی خطی با تاخیر زمانی

دانشجو: نفیسه رضوانی

استاد راهنما:

دکتر حجت احسنی طهرانی

استاد مشاور:

دکتر محمد مهدی فاتح

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

ماه و سال انتشار: اردیبهشت ۱۳۹۲

سپاس گذاری

از استاد گرامیم دکتر احسنی طهرانی به خاطر آموزش ها و راهنمایی های ایشان، صادقانه تشکر می کنم و از استاد مشاورم دکتر فاتح نیز به خاطر راهنمایی ایشان، بسیار متشکرم. همچنین از دکتر ناظمی و دکتر هاشمی که زحمت مطالعه و داوری این پایان نامه را کشیده اند، سپاس گزارم.

تعهد نامه

اینجانب نفیسه رضوانی دانشجوی دوره کارشناسی ارشد رشته ریاضی کاربردی دانشکده ریاضی دانشگاه صنعتی شاهرود نویسنده پایان نامه تخصیص مقدار ویژه برای سیستم های نوردای زمانی خطی با تاخیر زمانی تحت راهنمایی دکتر حجت احسنی طهرانی متعهد می شوم.

- تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهشهای محققان دیگر به مرجع مورد استفاده استناد شده است.
- مطالب مندرج در پایان نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است.
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و یا « Shahrood University of Technology » به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان نامه رعایت می گردد.
- در کلیه مراحل انجام این پایان نامه، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است.
- در کلیه مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری، ضوابط و اصول اخلاق انسانی رعایت شده است.

تاریخ

امضای دانشجو

مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج ، کتاب ، برنامه های رایانه ای ، نرم افزار ها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود می باشد. این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود .
- استفاده از اطلاعات و نتایج موجود در پایان نامه بدون ذکر مرجع مجاز نمی باشد.

چکیده

در این پایان نامه، روشی جدید که در سال های اخیر برای کنترل سیستم هایی از معادلات دیفرانسیل تاخیری و پایدارسازی آن ها با استفاده از تابع لامبرت مطرح شده را معرفی می کنیم. به این ترتیب روشی جدید برای طراحی کنترل کننده به وسیله تخصیص مقدار ویژه به همراه چند مثال ارائه می شود. با استفاده از این روش می توان یک زیر مجموعه از مقدارهای ویژه را به موقعیت های مطلوب انتقال داد. برای سیستمی که توسط معادلات دیفرانسیل تاخیری نشان داده می شود، جواب سیستم براساس تابع لامبرت بدست می آید و پایداری تعیین می شود. اگر سیستم پایدار نباشد، بعد از بررسی کنترل پذیری سیستم، یک پسخورد پایدار کننده به وسیله تخصیص مقدارهای ویژه طراحی می شود و سرانجام، سیستم حلقه بسته می تواند پایدار شود.

واژه های کلیدی: پایداری، تابع لامبرت، تاخیر زمانی، تخصیص مقدار ویژه، کنترل کننده پسخورد،

معادلات دیفرانسیل تاخیری

مقاله مستخرج از پایان نامه

[1] N. Rezvani, H. Ahsani Tehrani, "Control of linear systems with state and input delays," 4th Conference on Mathematical Analysis and its Applications, Faculty of Khansar, Khansar, Iran, May 7-8, 2013.

فهرست مطالب

| | | |
|-------|---|----|
| ۱ | فصل اول تعاریف و پیش نیازها | ۱ |
| ۱-۱ | تعاریف و پیش نیازهای ریاضی | ۱ |
| ۲-۱ | تعاریف و پیش نیازهای کنترل | ۳ |
| ۱-۲-۱ | حل معادلات فضای حالت | ۴ |
| ۲-۲-۱ | کنترل پذیری | ۵ |
| ۳-۲-۱ | پایداری سیستم های کنترل | ۵ |
| ۲ | فصل دوم مقدمه ای بر سیستم های تاخیری | ۸ |
| ۴-۲ | تاخیر | ۸ |
| ۴-۲ | معادلات دیفرانسیل تاخیری | ۹ |
| ۴-۲ | تابع لامبرت | ۱۵ |
| ۴-۲ | سیستم های تاخیری | ۱۷ |
| ۳ | فصل سوم بدست آوردن جواب سیستم های تاخیری به وسیله تابع لامبرت | ۲۰ |
| ۱-۳ | سیستم های همگن | ۲۱ |
| ۱-۱-۳ | حالت اسکالر | ۲۱ |
| ۲-۱-۳ | حالت ماتریسی | ۲۷ |
| ۲-۳ | سیستم های ناهمگن | ۳۳ |
| ۱-۲-۳ | حالت اسکالر | ۳۳ |
| ۲-۲-۳ | حالت ماتریسی | ۳۸ |

| | | |
|-------|--|----|
| ۴ | فصل چهارم بررسی پایداری سیستم های تاخیری به وسیله تابع لامبرت | ۴۱ |
| ۱-۴ | پایداری | ۴۱ |
| ۱-۱-۴ | حالت اسکالر | ۴۱ |
| ۲-۱-۴ | حالت ماتریسی | ۴۴ |
| ۵ | فصل پنجم تخصیص مقدار ویژه برای کنترل سیستم های تاخیری به وسیله تابع لامبرت | ۵۰ |
| ۱-۵ | تخصیص مقدار ویژه | ۵۱ |
| ۲-۵ | طراحی کنترل کننده پسخورد | ۵۴ |
| ۱-۲-۵ | حالت اسکالر | ۵۴ |
| ۲-۲-۵ | سیستم هایی با تاخیر در ورودی | ۶۱ |
| ۳-۲-۵ | سیستم هایی با تاخیر در متغیر حالت | ۶۶ |
| ۴-۲-۵ | سیستم هایی با تاخیر در متغیر حالت و ورودی | ۷۱ |
| ۶ | فصل ششم خلاصه و نتایج | ۷۵ |
| ۱-۶ | خلاصه | ۷۵ |
| ۲-۶ | نتایج | ۷۵ |
| ۳-۶ | مسئله های باز و کارهای آینده | ۷۷ |
| ۴-۶ | پیشنهادات | ۷۹ |
| ۷۹ | ضمیمه A (متن برنامه ها) | ۷۹ |
| ۸۸ | ضمیمه B (ضمیمه برای فصل پنجم) | ۸۸ |
| ۹۱ | مراجع | ۹۱ |

پیشگفتار

مسأله کنترل سیستم های تاخیری در چند دهه اخیر مورد توجه محققان زیادی قرار گرفته است به این خاطر که تاخیر زمانی اغلب در سیستم های متنوعی مانند مهندسی، پزشکی، شیمی، فیزیک، اقتصاد و ... اتفاق می افتد به طوریکه وجود تاخیر در این سیستم ها باعث عملکرد نامناسب آن ها می شود. این سیستم ها توسط معادلات دیفرانسیل تاخیری نشان داده می شوند که این معادلات به طور گسترده ای در دهه های گذشته مورد مطالعه قرار گرفته اند [۲۰]. وجود تاخیر باعث ایجاد تعداد نامتناهی ریشه در معادله مشخصه سیستم های تاخیری می شود به همین خاطر، تحلیل این سیستم ها با روش های کلاسیک مخصوصاً در بررسی پایداری و طراحی کنترل کننده پسخورد، مشکل می باشد. در طی سه دهه اخیر، مسأله تحلیل پایداری این سیستم ها مورد توجه قرار گرفته و مقالات زیادی در این زمینه مطرح شده است [۹]. همچنین روش هایی مانند روش SP^۱ در [۲۴] و روش FSA^۲ در [۳۳] با تبدیل مسأله به یک سیستم بدون تاخیر، برای کنترل سیستم های تاخیری استفاده شده است. چنین روش هایی ممکن است هنگامی که برای سیستم های واقعی به کار برده می شوند باعث خطاهای غیر منتظره شوند. بعلاوه اجرای کامل چنین روش هایی به خاطر مشکلات محاسباتی هنوز یک مسأله باز است. همچنین برای این سیستم ها کنترل کننده هایی با استفاده از توابع لیاپانوف^۳ در [۸، ۱۲] با به کار بردن نامعادلات ماتریسی خطی (LMIs)^۴ یا معادلات ریکاتی جبری (AREs)^۵ طراحی شده است. اگر چه می توان از چنین روش هایی برای انواع کلی تری از سیستم های تاخیری مانند سیستم های غیر خطی، سیستم های تغییر پذیر با زمان، سیستم هایی با تاخیرهای متغیر با زمان و سیستم های با چند تاخیر استفاده کرد اما این روش ها نیاز به فرمول بندی پیچیده ای دارند و از طرف دیگر روش های سیستماتیک کلی برای ساخت توابع لیاپانوف مناسب در دسترس نمی

^۱ Smith Predictor

^۲ Finite Spectrum Assignment

^۳ Lyapunov

^۴ Linear Matrix Inequalities

^۵ Algebraic Riccati Equations

باشد. اخیراً یک روش تحلیلی برای بدست آوردن جواب این سیستم ها و تحلیل آن ها با استفاده از تابع لامبرت در [۱، ۲۷] مطرح شده است که در این تحقیق به معرفی این روش می پردازیم و از آن برای کنترل چنین سیستم هایی استفاده می کنیم.

فصل اول از این پایان نامه، شامل مطالبی است که در فصل های بعد به آن ها نیاز داریم. در فصل دوم به معرفی تابع لامبرت می پردازیم و فرم فضای حالت سیستم هایی با تاخیر منحصر بفرد را نشان می دهیم. در فصل بعد به دنبال بدست آوردن جواب برای این سیستم ها با استفاده از تابع لامبرت می باشیم. در فصل چهارم نیز به تحلیل پایداری این سیستم ها با استفاده از این تابع می پردازیم. در فصل پنجم که فصل اصلی این پایان نامه می باشد ابتدا به بررسی پایداری سیستم کنترل پذیر می پردازیم و در صورتی که سیستم پایدار نباشد، با استفاده از تخصیص مقدار ویژه و با طراحی کنترل کننده پسخورد، سعی می کنیم سیستم را به صورت پایدار، کنترل نماییم. همچنین در ادامه این فصل از روش تابع لامبرت به عنوان روشی جدید برای کنترل سیستم های ناوردای زمانی خطی با تاخیر در متغیر حالت و ورودی استفاده می کنیم و سپس نتایج بدست آمده را با ارائه یک مثال نشان می دهیم. در فصل آخر نیز خلاصه و نتایج این تحقیق را بیان می کنیم.

فصل اول

تعاریف و پیش نیازها

۱-۱ تعاریف و پیش نیازهای ریاضی

تعریف ۱-۱ فرض کنیم A یک ماتریس $n \times n$ باشد. ریشه های چند جمله ای،

$$P_n(\lambda) = \det(A - \lambda I) \quad (1-1)$$

را که به چند جمله ای مشخصه ی ماتریس A موسوم است، مقادیر ویژه ی^۱ ماتریس A گوییم. به عبارت دیگر، λ را یک مقدار ویژه ی ماتریس A می نامیم، اگر و تنها اگر بردار غیر صفر x موجود باشد به طوری که،

$$Ax = \lambda x \quad (2-1)$$

بردار x را بردار ویژه ی^۲ متناظر با مقدار ویژه ی λ گوییم.

یکی از کاربردهای مهم مقادیر ویژه و بردارهای ویژه در قطری سازی ماتریس های مربعی است. اگر یک ماتریس $A_{n \times n}$ دارای n بردار ویژه مستقل خطی باشد، این ماتریس را با تبدیل تشابهی می توان قطری نمود ولی ماتریسی که مجموعه کاملی از n بردار ویژه مستقل خطی را نداشته باشد، نمی تواند قطری گردد چنین ماتریسی را باید به فرم متعارفی جردن^۳ تبدیل کرد.

قضیه زیر نشان می دهد که هر ماتریس دلخواه، همیشه مشابه با یک ماتریس بلوکی قطری می باشد.

قضیه ۱-۱ برای هر ماتریس دلخواه $A_{n \times n}$ ، یک ماتریس نامنفرد T وجود دارد به گونه ای که،

^۱Eigenvalue
^۲Eigenvector
^۳Jordan Canonical Form

$$J_{n \times n} = T^{-1}AT = \begin{bmatrix} J_1 & & & & \\ & J_2 & & & 0 \\ & & \ddots & & \\ & & & \ddots & \\ 0 & & & & J_m \end{bmatrix} \quad (3-1)$$

است که در آن هر یک از بلوک های جردن J_i به صورت،

$$J_i = \begin{bmatrix} \lambda_i & 1 & 0 & \dots & 0 \\ 0 & \lambda_i & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \lambda_i & 1 \\ 0 & 0 & \dots & 0 & \lambda_i \end{bmatrix}_{n_i \times n_i}, i = 1, \dots, m \quad (4-1)$$

می باشد که هر یک از λ_i ها مقادیر ویژه ی ماتریس A می باشند [5].

به ماتریس تبدیل T که ماتریس $A_{n \times n}$ را به فرم قطری بلوکی تبدیل می کند ماتریس مُدال^۱ نیز می گویند. در نرم افزار MATLAB از دستور $\text{jordan}(A)$ و $[T, J] = \text{jordan}(A)$ می توان برای بدست آوردن فرم متعارفی جردن ماتریس A استفاده نمود. دستور $\text{jordan}(A)$ فقط ماتریس متعارفی جردن حاصل را ارائه می دهد و در دستور $[T, J] = \text{jordan}(A)$ ماتریس T ماتریس تبدیل مربوطه و J ماتریس فرم متعارفی جردن ماتریس A است.

تعریف ۱-۲: ماتریس $A_{n \times n}$ را پوچتوان^۲ گویند هر گاه عدد طبیعی مانند m وجود داشته باشد که $A^m = 0$.

همچنین یادآوری می کنیم که برای محاسبه پوچی^۳ ماتریس A می توان رابطه زیر را به کار برد،

$$\text{rank}(A) + \text{nullity}(A) = n \quad (5-1)$$

که در آن $\text{rank}(A)$ رتبه ماتریس A را نشان می دهد.

^۱Matrix Modal

^۲Nilpotent

^۳Nullity

۲-۱ تعاریف و پیش نیازهای کنترل

در روش های مبتنی بر کنترل کلاسیک، مدل سازی و تحلیل سیستم ها براساس تابع تبدیل^۱ سیستم صورت می گیرد که این تابع ارتباط بین ورودی و خروجی سیستم را بیان می کند. اما در روش های مبتنی بر کنترل مدرن، مدل سازی سیستم بر پایه فضای حالت^۲ است. از جمله مزایای این روش مدل سازی، قابلیت استفاده از آن برای سیستم های چند ورودی- چند خروجی (MIMO)^۳، سیستم های خطی، غیر خطی و تغییر پذیر با زمان^۴ می باشد. صورت کلی معادلات فضای حالت به شکل زیر است،

$$\begin{aligned} \dot{x}(t) &= f[x(t), u(t), t] \\ y(t) &= g[x(t), u(t), t] \end{aligned} \quad (۶-۱)$$

که در آن $x_{n \times 1}(t)$ بردار حالت است و عناصر آن را متغیرهای حالت می نامند. $u_{m \times 1}(t)$ بردار ورودی و $y_{k \times 1}(t)$ بردار خروجی هستند. در حالت کلی، f و g نیز توابع غیر خطی متغیر با زمان هستند که نحوه ارتباط بردارهای حالت، ورودی و خروجی را نشان می دهند. برای سیستم های خطی و تغییر ناپذیر با زمان (ناوردای زمانی)^۵ معادلات به صورت زیر قابل ساده سازی هستند،

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad (۷-۱)$$

که در آن، $A_{n \times n}$ ماتریس حالت، $B_{n \times m}$ ماتریس ورودی، $C_{k \times n}$ ماتریس خروجی و $D_{k \times m}$ ماتریسی است که ارتباط مستقیم بین ورودی و خروجی را نشان می دهد.

متغیرهای حالت در یک سیستم می تواند تعبیر فیزیکی داشته باشد و قابل اندازه گیری با حسگر^۶ باشد مانند ولتاژ، جریان، دما، سرعت، فشار و جابجایی و نیز می تواند کاملاً ریاضی باشد و تعبیر

^۱Transfer Function

^۲State Space

^۳Multi Input-Multi Output

^۴Time Varying

^۵Time Invariant

^۶Sensor

فیزیکی نداشته باشد و علت استفاده از آن ها فقط برای ساده سازی محاسبات ریاضی است. از آنجایی که نمایش فضای حالت، بستگی به انتخاب متغیرهای حالت انتخاب شده دارد به همین دلیل، برخلاف تابع تبدیل که یک نمایش منحصر به فرد از یک سیستم است، نمایش های فضای حالت متعددی برای یک سیستم می توان بدست آورد.

۱-۲-۱ حل معادلات فضای حالت

صورت کلی معادلات فضای حالت را در نظر بگیرید،

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

برای حل این معادلات لازم است تا بردار $x(t)$ را از معادله اول بدست آورد و با جایگذاری آن ها در معادله دوم، خروجی سیستم را محاسبه نمود. معادله اول را در نظر بگیرید،

$$\dot{x}(t) = Ax(t) + Bu(t)$$

پاسخ کلی چنین معادله ای به صورت زیر می باشد،

$$x(t) = e^{A(t-t_0)}x(t_0) + e^{At} \int_{t_0}^t e^{-A\tau} Bu(\tau) d\tau \quad (۸-۱)$$

e^{At} تابع نمایی ماتریسی است که به آن ماتریس انتقال حالت^۱ می گویند و با نماد $\varphi(t)$ نیز نمایش می دهند،

$$x(t) = \varphi(t - t_0)x(t_0) + \int_{t_0}^t \varphi(t - \tau)Bu(\tau) d\tau$$

در صورتی که معادلات را به صورت همگن در نظر بگیریم،

$$\dot{x}(t) = Ax(t)$$

در این صورت، جواب معادله همگن به شکل زیر خواهد بود،

^۱State Transition Matrix

$$x(t) = e^{A(t-t_0)}x(t_0) = \varphi(t - t_0)x(t_0) \quad (9-1)$$

که در آن ماتریس انتقال حالت در واقع بیان کننده پاسخ طبیعی یا بدون ورودی سیستم می باشد.

۲-۲-۱ کنترل پذیری

تعریف ۳-۱ سیستم خطی ناوردای زمانی،

$$\dot{x}(t) = Ax(t) + Bu(t)$$

را به ازای $t \geq t_0$ و با حالت اولیه ی $x(t_0) = x_0$ در نظر می گیریم. هرگاه ورودی $u(t)$ که $t \in [t_0, t_1]$ ، حالت اولیه ی x_0 را در زمان t_1 به حالت صفر انتقال دهد، حالت x_0 را در زمان t_0 کنترل پذیر^۱ گوییم. لازم به ذکر است حالت صفر در اینجا بیانگر حالت تعادل می باشد.

کنترل پذیری یک سیستم، اهمیتی اساسی دارد. زیرا باید مسائلی مورد مطالعه قرار گیرند که در آن ها هدف، انتقال سیستم از یک حالت اولیه ی دلخواه، به حالت تعادل می باشد. در نتیجه کنترل پذیری، شرط لازم برای وجود جواب است.

۳-۲-۱ پایداری سیستم های کنترل

در طراحی یک سیستم کنترل، شناخت اجزای سیستم برای پیش بینی رفتار دینامیکی سیستم لازم است. یک سیستم کنترل خطی ناوردای زمانی، در صورتی پایدار است که هنگام اعمال یک شرط اولیه جدید به آن، به حالت تعادل خود برگردد. حالت تعادل برای یک سیستم کنترل خطی، حالتی است که در صورت نبود ورودی و اغتشاش، خروجی در آن حالت باقی بماند.

حال به بررسی مفهوم ریاضی پایداری در سیستم های کنترل می پردازیم. سیستم کنترل خطی ناوردای زمانی،

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (10-1)$$

^۱Controllable

با بردار ورودی،

$$u(t) = Kx(t) \quad (11-1)$$

موسوم به قانون کنترل، که در آن $u(t)$ ، متناسب با بردار حالت $x(t)$ انتخاب شده است را در نظر می گیریم. در این صورت، سیستم کنترل را یک سیستم کنترل با پس خورد حالت و $K \in \mathcal{R}^{m \times n}$ را ماتریس پسخورد حالت^۱ گوییم.

با ترکیب دو معادله (۱۰-۱) و (۱۱-۱) رابطه ی زیر بدست می آید،

$$\dot{x}(t) = (A + BK)x(t) \quad (12-1)$$

در این رابطه ماتریس A را ماتریس حلقه باز و مقادیر ویژه آن را مقادیر ویژه سیستم حلقه باز گوییم. همچنین ماتریس $A + BK$ را ماتریس سیستم حلقه بسته و مقادیر ویژه آن را مقادیر ویژه سیستم حلقه بسته می نامیم.

در ادامه، برخی از تعاریف و اصطلاحات را که در مطالعه سیستم های کنترل با آنها برخورد خواهیم داشت، ارائه می کنیم.

تعریف ۴-۱ سیستم هایی که در آن ها خروجی هیچ نقشی بروی عمل کنترل ندارد را سیستم های کنترل حلقه باز گویند [۳۴].

تعریف ۵-۱ سیستم های کنترل پس خوردی را عموماً سیستم های کنترل حلقه بسته می نامیم. منظور از کنترل حلقه بسته استفاده از پس خورد برای کاهش خطای سیستم و رسیدن به پایداری است.

قضیه ۲-۱ فرض کنید A یک ماتریس $n \times n$ باشد. به ازای هر بردار حالت اولیه ی x_0 ، سیستم خطی

^۱State Feedback Matrix

ناوردای زمانی،

$$\dot{x}(t) = Ax(t), x(0) = x_0$$

دارای یک جواب منحصر به فرد به صورت زیر می باشد،

$$x(t) = e^{At} x_0$$

با استفاده از این قضیه و انتخاب بردار حالت اولیه ی x_0 ، جواب سیستم حلقه بسته (۱۰-۱) و (۱۱-۱)

را به صورت زیر بدست می آوریم،

$$x(t) = e^{(A+BK)t} x_0$$

چنانچه تمام مقادیر ویژه سیستم حلقه بسته، در سمت چپ صفحه مختلط قرار گیرند آنگاه داریم،

$$\lim_{t \rightarrow \infty} x(t) = x_0 \lim_{t \rightarrow \infty} e^{(A+BK)t} = 0$$

این رابطه بیان می کند که با گذر زمان، هر حالت اولیه در صورتی به حالت تعادل برده می شود که

قسمت حقیقی تمام مقادیر ویژه ماتریس حلقه بسته، منفی باشند.

تعریف ۱-۶ سیستم کنترل خطی تعریف شده توسط معادلات (۱۰-۱) و (۱۱-۱) را یک سیستم پایدار

مجانبی می نامیم هر گاه قسمت حقیقی همه مقادیر ویژه ماتریس حلقه بسته منفی باشد. در صورتی

که قسمت حقیقی همه مقادیر ویژه منفی یا صفر باشد سیستم را پایدار و در غیر این دو صورت

سیستم را ناپایدار نامیم.

تعریف ۱-۷ مسأله یافتن ماتریس پسخورد K ، برای سیستم کنترل خطی تعریف شده توسط

معادلات (۱۰-۱) و (۱۱-۱) به گونه ای که سیستم پایدار مجانبی باشد را مسأله تخصیص مقادیر ویژه

گوییم.

در تحلیل و طراحی های این پایان نامه تنها سیستم های خطی و تغییر ناپذیر با زمان در نظر گرفته

خواهند شد.

فصل دوم

مقدمه ای بر سیستم های تاخیری

سیستم های تاخیری در دنیا فراوان هستند. یک دلیل آن است که طبیعت پر از تاخیرهای آشکار است. دلیل دیگر آن است که سیستم های تاخیری غالباً به عنوان مدلی برای یک رده بزرگ از سیستم های مهندسی به کار برده می شوند که در آن انتشار و انتقال اطلاعات یا ماده مورد بحث است. وجود تاخیر مخصوصاً تاخیرهای طولانی باعث می شود آنالیز و کنترل این سیستم ها در مقایسه با سیستم های بدون تاخیر بسیار پیچیده تر شود. در این فصل، چند مثال از سیستم های تاخیری مطرح می کنیم و همچنین به معرفی تابع لامبرت می پردازیم و فرم فضای حالت این سیستم ها را نشان می دهیم.

۱-۲ تاخیر

تاخیر زمانی^۱ ویژگی یک سیستم فیزیکی است که در نتیجه آن، واکنش یا پاسخ سیستم به یک نیروی وارد شده، به تاخیر می افتد. هر زمان که اطلاعات یا انرژی به لحاظ فیزیکی از یک مکان به مکان دیگر انتقال داده می شود به همراه انتقال، تاخیر به وجود می آید که مقدار آن به فاصله و سرعت انتقال بستگی دارد به این ترتیب بعضی تاخیرها کوتاه و برخی دیگر خیلی طولانی هستند.

تاخیر زمانی در هر یک از اجزای تشکیل دهنده ساختار یک سیستم می تواند به وجود آید و در سیستم های متنوعی مانند سیستم های بیولوژیکی، زیستی، صنعتی، الکتریکی، هوایی، اقتصادی، اجتماعی، فرایندهای شیمیایی، شبکه های هیدرولیکی و شبکه های ارتباطی و ... اتفاق می افتد. وجود تاخیر مخصوصاً تاخیرهای طولانی در این سیستم ها گاهی باعث ناپایداری می شود به همین خاطر در طی سه دهه اخیر، تحلیل این سیستم ها و مسأله پایداری آن ها موضوعاتی هستند که مورد توجه

^۱Time Delay

محققان قرار گرفته اند.

۲-۲ معادلات دیفرانسیل تاخیری

معادلات دیفرانسیل تاخیری (DDEs)^۱ یک رده مهم و بزرگ از سیستم های دینامیکی هستند که در قرن هجدهم توسط کاندورست^۲ و لاپلاس^۳ معرفی شده اند [۹].

از آنجایی که معادلات دیفرانسیل معمولی (ODEs)^۴ به صورت،

$$\dot{x}(t) = f(t, x(t)) \quad (1-2)$$

یک مدل رایج برای نمایش سیستم های دینامیکی هستند که در این مدل، متغیرهای $x(t) \in \mathcal{R}^n$ متغیرهای حالت می باشند و معادله دیفرانسیل، تغییر متغیر های حالت را نسبت به زمان مشخص می کند، اما برای نمایش برخی از سیستم های دینامیکی مانند سیستم های تاخیری نمی توان از این معادلات استفاده کرد زیرا در این سیستم ها برخلاف سیستم های بدون تاخیر، رفتار سیستم نه تنها به حالت های فعلی بلکه همچنین به حالت های قبل نیز وابسته است.

برای نمایش این سیستم ها همانطور که اشاره کردیم از معادلات دیفرانسیلی استفاده می شود که در آنها فاکتورهای تاخیر زمانی وجود دارد که به آنها معادلات دیفرانسیل تاخیری می گویند.

صورت کلی معادله دیفرانسیل تاخیری برای $x(t) \in \mathcal{R}^n$ به صورت،

$$\dot{x}(t) = f(t, x(t), x_t) \quad (2-2)$$

می باشد که در آن $x_t = \{x(h): h \leq t\}$ و f یک تابع از $\mathcal{R} \times \mathcal{R}^n \times \mathcal{C}$ به \mathcal{R}^n است. نمایشی که در این پایان نامه برای نشان دادن یک سیستم با تاخیر منحصر به فرد استفاده می شود به صورت،

^۱Delay Differential Equations

^۲Condorcet

^۳Laplace

^۴Ordinary Differential Equations

$$\dot{x}(t) = f(t, x(t), x(t-h)) \quad (3-2)$$

است که در آن f یک تابع خطی است و h تاخیر زمانی را نشان می دهد. وجود تاخیر در این معادله باعث ایجاد عبارت نمایی در معادله مشخصه سیستم های تاخیری می شود [۳۲] به همین دلیل، معادله مشخصه این سیستم ها دارای تعداد نامتناهی ریشه می باشد. در واقع، مشکل اصلی این معادلات آن است که باعث ایجاد یک طیف نامتناهی از مقادیرهای ویژه می شوند. بنابراین تحلیل این سیستم ها با روش های کلاسیک به خصوص در بررسی پایداری در مقایسه با سیستم های بدون تاخیر، پیچیده تر خواهد بود.

چنین مسائلی اغلب به طور غیر مستقیم و با استفاده از روش های ترسیمی، عددی و تقریبی حل می شوند. در بسیاری موارد از روش تقریب پاده^۱ استفاده می شود که یک تقریب گویا است و در این روش به جای عبارت نمایی در معادله مشخصه، یک عبارت کسری جایگزین می شود. اما این روش شامل محدودیت هایی است و می تواند منجر به ناپایداری در سیستم اصلی شود [۹]. روش های دیگری نیز وجود دارد که از آن ها برای تبدیل سیستم تاخیری به یک سیستم بدون تاخیر و سپس پایدار کردن آن استفاده می شود. چنین روش هایی ممکن است زمانی که برای یک سیستم واقعی به کار برده می شوند، باعث خطاهای غیر منتظره شوند. علاوه بر این اجرای مطمئن آنها هنوز یک مسأله باز است.

در سال های اخیر، استفاده از روش تابع لامبرت^۲ در زمینه های علمی متنوع و مهندسی شامل تحلیل سیستم های تاخیری رواج پیدا کرده است. ایده به کار بردن تابع لامبرت برای بررسی سیستم های تاخیری متعلق به رایت^۳ می باشد [۲۵]. همچنین این مسأله توسط محققان دیگر نیز مورد بررسی قرار گرفته و بهبود داده شده است [۱، ۲]. روشی که در این پایان نامه برای پایداری سازی سیستم های تاخیری و کنترل آن ها به کار برده می شود، استفاده از تابع لامبرت می باشد که روشی جدید برای

^۱Pade Approximation

^۲Lambert W Function

^۳Wright

آنالیز و کنترل سیستم های تاخیری است.

قبل از معرفی تابع لامبرت و برای درک این مطلب که چگونه تاخیر زمانی در عمل اتفاق می افتد، به بیان چند مثال ساده از سیستم های تاخیری می پردازیم.

مثال ۱-۲ (سیستم هیدرولیکی) در شکل ۱-۲ یک مخزن آب و یک لوله به طول l نشان داده شده است. برای این مخزن آب یک مدل خطی به صورت زیر در نظر گرفته شده است [۶]،

$$\dot{x}(t) = \frac{1}{A\rho} \left(u_{in}(t-h) - \frac{\rho g x(t)}{R} \right)$$

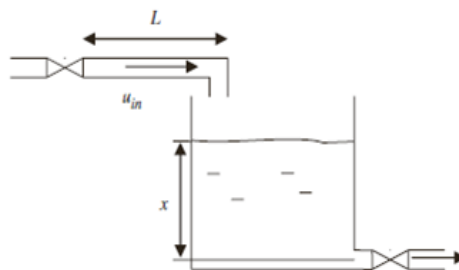
که در آن A سطح مقطع عرضی است، ρ چگالی آب، g شتاب گرانشی، h تاخیر زمانی، t زمان، R مقاومت هیدرودینامیکی لوله خروجی، u_{in} سرعت جریان ورودی و $x(t)$ سطح تراز آب می باشد. هدف از کنترل آن است که سطح آب درون مخزن $0.5m$ باشد. بنابراین با در نظر گرفتن یک کنترل پسخورده^۱ مناسب به صورت،

$$u_{in} = kx(t)$$

سیستم حلقه بسته می تواند توسط یک معادله دیفرانسیل تاخیری و به صورت،

$$\dot{x}(t) = -\frac{g}{AR}x(t) + \frac{k}{A\rho}x(t-h)$$

نشان داده شود.



شکل ۱-۲: سیستم هیدرولیکی

مثال ۲-۲ (مسأله آونگ) معادله حرکت یک آونگ ساده را به صورت زیر در نظر بگیرید،

$$T(t-h) - mgl\sin(\theta(t)) = ml^2\ddot{\theta}(t)$$

که در آن l طول آونگ، g شتاب گرانشی، m جرم آونگ و T گشتاور ورودی می باشد.

هدف نگه داشتن آونگ در نقطه $\theta = \theta_0$ است. می توان نشان داد که مدل غیر خطی می تواند در

اطراف $\theta = \theta_0$ به صورت خطی زیر تبدیل شود،

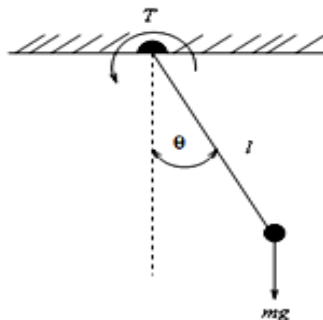
$$\begin{bmatrix} \Delta\dot{\theta}(t) \\ \Delta\ddot{\theta}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l}\cos(\theta_0) & 0 \end{bmatrix} \begin{bmatrix} \Delta\theta(t) \\ \Delta\dot{\theta}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{g\sin(\theta_0)}{l} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} T(t-h)$$

با در نظر گرفتن $\Delta\theta = \theta - \theta_0$ و

$$T(t) = [k_1 \quad k_2] \begin{bmatrix} \Delta\theta(t) \\ \Delta\dot{\theta}(t) \end{bmatrix} + mgl\sin(\theta_0)$$

بنابراین سیستم حلقه بسته می تواند به صورت زیر نشان داده شود،

$$\begin{bmatrix} \Delta\dot{\theta}(t) \\ \Delta\ddot{\theta}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l}\cos(\theta_0) & 0 \end{bmatrix} \begin{bmatrix} \Delta\theta(t) \\ \Delta\dot{\theta}(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{k_1}{ml^2} & \frac{k_2}{ml^2} \end{bmatrix} \begin{bmatrix} \Delta\theta(t-h) \\ \Delta\dot{\theta}(t-h) \end{bmatrix}$$



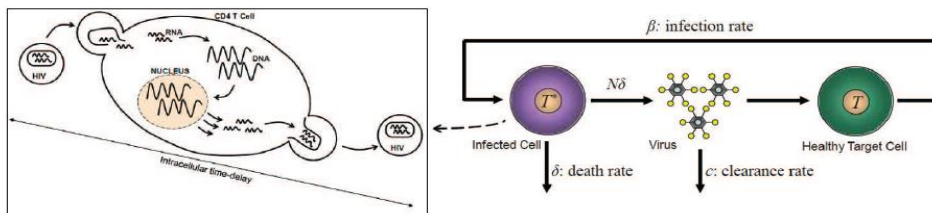
شکل ۲-۲: آونگ ساده

مثال ۳-۲ (مدل سازی بیماری HIV با یک تاخیر درون سلولی) سالها تلاش شده است تا

براساس معادلات دیفرانسیل، مدلی برای دینامیک بیماری HIV بدست آید. چنین مدلهایی می توانند

در تحلیل و آنالیز بهتر بیماری HIV و طراحی رژیم های درمانی مناسب کمک کنند. همانطور که در

شکل ۳-۲ می بینید HIV ، سلول های هدف^۱ را با یک سرعت خاص، آلوده می کند. در مقابل داروهایی مانند بازدارنده پروتئاز^۲ به جلوگیری از تقسیم شدن و انتقال پلی پروتئین های ویروسی کمک می کنند. سیستم ایمنی بدن انسان نیز برخی از ویروس های عفونی را از بین می برد. چون زمانی طول می کشد تا ویروس وارد یک سلول هدف شود و قبل از تولید ویروس جدید، یک تاخیر (تاخیر درون سلولی) در این فرایند زیستی اتفاق می افتد.



شکل ۳-۲: HIV با سرعت ثابت β سلول های هدف T را به مولد سلول های آلوده T^* تبدیل می کند [۳۲]

همانطور که در شکل ۳-۲ مشاهده می شود HIV با یک سرعت ثابت، سلول های هدف T را آلوده می کند و باعث می شود آن ها به تولید کننده سلول های آلوده T^* تبدیل شوند. زمان تاخیر از زمان بین ورود ویروس اولیه به یک سلول و تولید ویروس بعدی حاصل می شود.

این مدل می تواند به وسیله معادلات دیفرانسیل تاخیری و به صورت زیر مدل سازی شود [۳۲]،

$$\frac{dT^*(t)}{dt} = \beta T_0 V_I(t-h) - \delta T^*(t)$$

$$\frac{dV_I(t)}{dt} = (1 - n_p) N \delta T^*(t) - c V_I(t)$$

$$\frac{dV_{NI}(t)}{dt} = n_p N \delta T^*(t) - c V_{NI}(t)$$

که در آن t زمان سپری شده، h زمان تاخیر، T_0 غلظت (تراکم) سلول هدف، T^* غلظت مولد سلول آلوده، V_I غلظت ویروس های عفونی در پلاسما، V_{NI} غلظت ویروس های غیر عفونی در پلاسما، c سرعت از بین رفتن ویروس، δ سرعت از بین رفتن سلول تولید کننده ویروس، N متوسط تعداد

^۱Target Cells

^۲Protease

ویروس های جدید مولد هر سلول آلوده در طی مدت عمر و n_p اثر داروی بازدارنده پروتئاز را نشان می دهد. همچنین در این مدل فرض شده است که HIV سلول های هدف را با سرعت β آلوده می کند و باعث می شود آن ها به مولد سلول های عفونی تبدیل شوند. این مدل همچنین می تواند به وسیله معادلات دیفرانسیل تاخیری ماتریسی و به صورت زیر نشان داده شود،

$$\begin{bmatrix} \dot{T}^*(t) \\ \dot{V}_I(t) \\ \dot{V}_{NI}(t) \end{bmatrix} = \begin{bmatrix} -\delta & 0 & 0 \\ (1-n_p)N\delta & -c & 0 \\ n_p N\delta & 0 & -c \end{bmatrix} \begin{bmatrix} T^*(t) \\ V_I(t) \\ V_{NI}(t) \end{bmatrix} + \begin{bmatrix} 0 & \beta T_0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} x(t-h), \quad t > 0$$

که در آن،

$$x(t) = g(t) = \{T_{ss}^* \quad V_{ss} \quad 0\}^T, t \in [-h, 0]$$

می باشد.

یک مثال ساده دیگر از یک سیستم تاخیری که در زندگی روزمره اتفاق می افتد کنترل دما است. همانطور که در شکل ۲-۴ نشان داده می شود اکثر افراد مشکل تنظیم دمای آب را تجربه کرده اند دمای آب گاهی اوقات خیلی داغ یا خیلی سرد است. دمای واقعی اغلب خارج از حد مطلوب است و گاهی مدتی طول می کشد تا دما کنترل شود و این به خاطر آن است که برای جاری شدن آب از شیر به سمت بدن انسان زمانی صرف می شود. این زمان یک تاخیر است که به فشار آب و طول لوله بستگی دارد.

در این مثال با فرض این که آب یک مایع تراکم ناپذیر است و براساس قانون پویزیول^۱ سرعت جریان آب برابر است با [۳۳]،

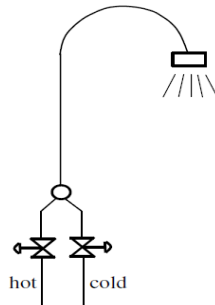
$$F = \frac{\pi R^4}{8\mu l} \Delta p$$

که در آن μ برابر 0.01 چسبندگی جنبشی آب را نشان می دهد. همچنین R شعاع لوله، l طول لوله و

^۱Poiseuille

Δp تفاضل فشار دو سر لوله را نشان می دهند. تاخیر زمانی h را می توان به صورت زیر بدست آورد،

$$h = \frac{\pi R^2 l}{F} = \frac{8\mu}{\Delta p} \left(\frac{l}{R}\right)^2$$



شکل ۲-۴: طرح ساده از سیستم دوش

۲-۳ تابع لامبرت

این تابع توسط لامبرت و اوپلر در سال های ۱۷۲۸ تا ۱۷۸۳ معرفی شده است [۴] و در ریاضیات به آن تابع آمگا یا ضرب لگاریتمی می گویند، به طوریکه هر تابع $\omega = W(H)$ که در رابطه زیر صدق می کند،

$$W(H)e^{W(H)} = H \quad (۴-۲)$$

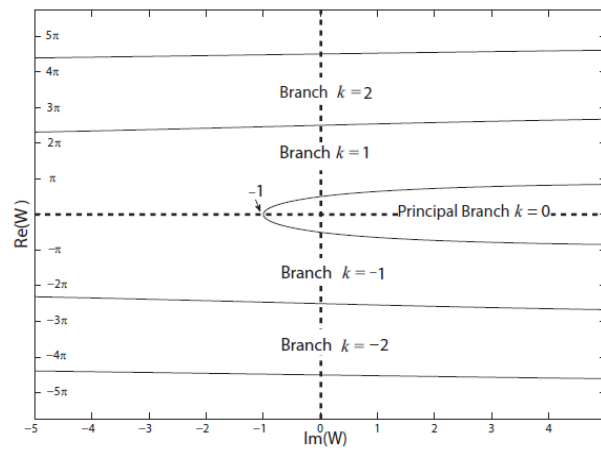
تابع لامبرت نامیده می شود. این تابع، مختلط مقدار با یک آرگومان مختلط H و دارای تعداد نامتناهی شاخه است به طوریکه هر شاخه با یک اندیس k به صورت W_k و برای $k = -\infty, \dots, -1, 0, 1, \dots, \infty$ نشان داده می شود که در آن W_0 ، شاخه اصلی تابع لامبرت نامیده می شود و هر یک از شاخه ها در رابطه،

$$W_k(H)e^{W_k(H)} = H \quad (۵-۲)$$

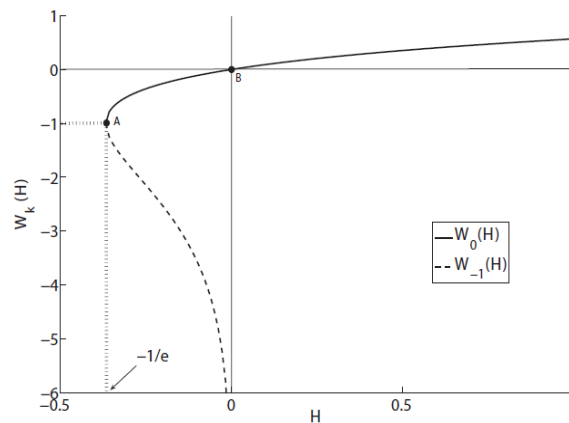
صدق می کند [۲۶]. شاخه اصلی و شاخه های دیگر از تابع لامبرت می توانند با استفاده از توابع موجود در نرم افزارهایی مانند MATLAB محاسبه شوند. در این پایان نامه نیز از این نرم افزار برای محاسبات استفاده می کنیم.

همانطور که در شکل ۵-۲ مشاهده می شود هر شاخه $W_k(H)$ از تابع لامبرت که در آن H یک مقدار مختلط است، دارای تصویر مخصوص به خود می باشد. همچنین تصویر هر شاخه نسبت به محور حقیقی متقارن است. قسمت حقیقی شاخه اصلی نیز دارای مینیمم مقدار -1 در نقطه $H = \frac{-1}{e}$ است (شکل ۶-۲ را ببینید). به بیان دیگر داریم [۴]،

$$\text{Re}\{W_0(H)\} \geq -1 \quad (6-2)$$



شکل ۵-۲: تصاویر شاخه های تابع لامبرت [۴]



شکل ۶-۲: مقادیر حقیقی شاخه های $k = -1$ و $k = 0$ از تابع لامبرت [۴]

یکی دیگر از خصوصیت های تابع لامبرت که از آن در تعیین راست ترین مقادیرهای ویژه^۱ و سپس

^۱Rightmost Eigenvalues

تعیین پایداری سیستم تاخیری استفاده می شود به صورت زیر می باشد [۲۷]،

$$\max[\operatorname{Re}\{W_k(H)\}] = \operatorname{Re}\{W_0(H)\} \quad (۷-۲)$$

همچنین از دیگر خواص مهم تابع لامبرت، استفاده از آن برای حل معادلات شامل عبارت های نمایی می باشد. برای مثال به منظور حل معادله زیر با استفاده از تابع لامبرت به صورت زیر عمل می کنیم،

$$2^t = 3t + 1 \rightarrow 1 = (3t + 1)e^{-t \ln 2} \rightarrow \frac{1}{3} = \left(t + \frac{1}{3}\right) e^{-t \ln 2} \rightarrow -\frac{\ln 2}{3} = -\left(t \ln 2 + \frac{\ln 2}{3}\right) e^{-t \ln 2}$$

$$\rightarrow -\frac{\ln 2}{3} e^{-\frac{\ln 2}{3}} = -\left(t \ln 2 + \frac{\ln 2}{3}\right) e^{-t \ln 2 - \frac{\ln 2}{3}} \rightarrow -\left(t \ln 2 + \frac{\ln 2}{3}\right) = W\left(-\frac{\ln 2}{3} 2^{-\frac{1}{3}}\right)$$

$$\rightarrow t = -\frac{1}{\ln 2} W\left(-\frac{\ln 2}{3} 2^{-\frac{1}{3}}\right) + \frac{1}{3}$$

به این ترتیب جواب معادله با استفاده از تابع لامبرت بدست می آید.

بنابراین می توان گفت مزیت اصلی استفاده از تابع لامبرت این است که به کمک آن می توان معادله مشخصه سیستم های تاخیری خطی اسکالر را حل کرد و سپس به کمک نرم افزار MATLAB که دارای یک تابع به نام lambertw برای محاسبه تابع لامبرت است، موقعیت ریشه ها را مشخص کرد.

تابع لامبرت را می توان با استفاده از روش نیوتن و با تقریب های متوالی برای $\omega = W(H)$ به صورت زیر تقریب زد،

$$\omega_{j+1} = \omega_j - \frac{\omega_j e^{\omega_j} - H}{e^{\omega_j} + \omega_j e^{\omega_j}} \quad (۸-۲)$$

مشتق از تابع لامبرت نیز از رابطه زیر محاسبه می شود،

$$\frac{dW}{dH} = \frac{W(H)}{H(1+W(H))}, H \neq 0, -1/e \quad (۹-۲)$$

۴-۲ سیستم های تاخیری

همانطور که قبلاً نیز بیان کردیم سیستم های خطی شامل تاخیر زمانی به وسیله معادلات دیفرانسیل

تاخیری نشان داده می شوند به طوریکه معادله فضای حالت یک سیستم نوردای زمانی خطی با تاخیر منحصر به فرد به صورت زیر می باشد،

$$\dot{x}(t) = Ax(t) + A_d x(t-h), t > 0 \quad (10-2)$$

که در آن A و A_d ماتریس های $n \times n$ هستند، $x(t)$ یک بردار حالت $n \times 1$ است و $h > 0$ تاخیر زمانی را نشان می دهد. برای این که معادله (10-2) جواب یکتا داشته باشد یک تابع برداری اولیه $g(t)$ که $n \times 1$ نیز می باشد، باید وجود داشته باشد به گونه ای که،

$$x(t) = g(t), t \in [-h, 0] \quad (11-2)$$

برای حل معادله (10-2) می توان از تبدیل لاپلاس استفاده کرد و با توجه به این که،

$$\mathcal{L}(x(t-h)) = \int_0^{\infty} e^{-st} x(t-h) dt = \int_0^h e^{-st} x(t-h) dt + \int_h^{\infty} e^{-st} x(t-h) dt$$

که در آن \mathcal{L} عملگر لاپلاس می باشد و همچنین از این که،

$$0 < t < h \rightarrow -h < t-h < 0 \rightarrow x(t-h) = g(t-h)$$

بدست می آوریم،

$$\begin{aligned} \mathcal{L}(x(t-h)) &= \int_0^h e^{-st} g(t-h) dt + \int_0^{\infty} e^{-s(t+h)} x(t) dt = \int_0^h e^{-st} g(t-h) dt \\ &+ e^{-sh} \int_0^{\infty} e^{-st} x(t) dt \end{aligned}$$

با توجه به اینکه،

$$X(s) = \mathcal{L}(x(t)) = \int_0^{\infty} e^{-st} x(t) dt$$

و با در نظر گرفتن $x(0) = x_0$ خواهیم داشت،

$$sX(s) - x_0 = AX(s) + A_d \left(e^{-sh} X(s) + \int_0^h e^{-st} g(t-h) dt \right) \quad (12-2)$$

با ساده کردن رابطه (12-2) بدست می آوریم،

$$(sI - A - A_d e^{-sh})X(s) = x_0 + A_d \int_0^h e^{-st} g(t-h) dt, \quad (13-2)$$

بنابراین داریم [۲۱]،

$$X(s) = (sI - A - A_d e^{-sh})^{-1} \left(x_0 + A_d \int_0^h e^{-st} g(t-h) dt \right) \quad (14-2)$$

به این ترتیب جواب سیستم می تواند با استفاده از عکس تبدیل لاپلاس و از رابطه (۱۳-۲) محاسبه شود. معادله مشخصه سیستم نیز با فرض یک جواب غیر صفر به فرم $e^{st}C$ که در آن s یک متغیر اسکالر و C یک ثابت است به صورت،

$$\det(sI - A - A_d e^{-sh}) = s^n + a_{n-1}(e^{-sh})s^{n-1} + \dots + a_1(e^{-sh})s + a_0(e^{-sh}) = 0 \quad (15-2)$$

بدست می آید که در آن s ریشه های معادله مشخصه یا همان مقادیر ویژه می باشند و a_i ها برای $i = 1, \dots, n-1$ توابعی از e^{-sh} هستند [۹]. واضح است که این معادله دارای تعداد نامتناهی جواب است. برای تحلیل پایداری یک سیستم تاخیری می توان از این معادله مشخصه استفاده کرد که در فصل چهارم به بررسی آن می پردازیم.

فصل سوم

بدست آوردن جواب سیستم های تاخیری به وسیله تابع لامبرت

تاکنون تلاش های زیادی برای پیدا کردن یک جواب تحلیلی برای معادلات دیفرانسیل تاخیری به وسیله حل معادله مشخصه تحت شرایط متفاوت انجام شده است. وجود و یکتایی جواب برای معادلات دیفرانسیل تاخیری خطی توسط هال^۱ و لائل^۲ اثبات شده است. همچنین یکتایی جواب توسط رایت نیز مورد مطالعه قرار گرفته است. اخیراً یک روش برای حل سیستم های نوردای زمانی خطی از معادلات دیفرانسیل تاخیری با استفاده از تابع لامبرت توسط آسل^۳ و آلسوی^۴ مطرح شده است. برخلاف نتایج بدست آمده از سایر روش ها جواب دارای فرم تحلیلی است و از فرم جواب بدست آمده، می توان تشخیص داد که چگونه پارامترها در آن قرار می گیرند و روی مقدارهای ویژه و جواب سیستم، تأثیر می گذارند همچنین هر مقدار ویژه متناظر با یک شاخه از تابع لامبرت است. به این ترتیب مفهوم ماتریس انتقال حالت در معادلات دیفرانسیل معمولی می تواند با استفاده از ماتریس تابع لامبرت برای معادلات دیفرانسیل تاخیری تعمیم داده شود. در این فصل، از روش مطرح شده برای حل سیستم هایی از معادلات دیفرانسیل تاخیری و برای حالت اسکالر و برخی حالت های خاص دیگر استفاده می کنیم و روش را برای حالت های کلی تر که در آن ماتریس های ضرایب سیستم از معادلات دیفرانسیل تاخیری جابجا نمی شوند و حالت ناهمگن نیز به کار می بریم.

^۱Hale

^۲Lunel

^۳Asl

^۴Ulsoy

۱-۳ سیستم های همگن

۱-۱-۳ حالت اسکالر

معادله دیفرانسیل تاخیری همگن و از مرتبه اول زیر را در نظر می گیریم،

$$\begin{aligned} \dot{x}(t) + ax(t) + a_d x(t-h) &= 0, h > 0, t > 0 \\ x(t) &= g(t), t \in [-h, 0] \end{aligned} \quad (1-3)$$

با فرض اینکه جواب معادله به صورت،

$$x(t) = e^{st} C^I \quad (2-3)$$

باشد که در آن s یک عدد مختلط و C^I نیز هر عدد مختلط قابل قبول است و با قرار دادن $x(t)$ در

معادله سیستم بدست می آوریم،

$$(se^{st} + ae^{st} + a_d e^{s(t-h)})C^I = 0 \rightarrow (s + a + a_d e^{-sh})e^{st}C^I = 0 \quad (3-3)$$

چون $e^{st}C^I \neq 0$ آنگاه نتیجه می گیریم،

$$(s + a + a_d e^{-sh}) = 0 \quad (4-3)$$

با ضرب رابطه (۴-۳) در e^{sh} خواهیم داشت،

$$(s + a)e^{sh} = -a_d \quad (5-3)$$

همچنین رابطه (۵-۳) را در he^{ah} ضرب کرده و بدست می آوریم،

$$(s + a)he^{(s+a)h} = -a_d he^{ah} \quad (6-3)$$

همانطور که می دانیم هر تابع $W(H)$ که در رابطه $W(H)e^{W(H)} = H$ صدق می کند یک تابع لامبرت

است. بنابراین داریم،

$$W(-a_d he^{ah}) = (s + a)h \quad (7-3)$$

از رابطه (۷-۳) نتیجه می گیریم،

$$s = \frac{1}{h} W(-a_d h e^{ah}) - a \quad (۸-۳)$$

و همچنین بدست می آوریم،

$$W(-a_d h e^{ah}) e^{W(-a_d h e^{ah})} = -a_d h e^{ah} \quad (۹-۳)$$

به طوریکه $W(-a_d h e^{ah})$ مقدار تابع لامبرت را در $H = -a_d h e^{ah}$ نشان می دهد.

به این ترتیب جواب کلی معادله دیفرانسیل مرتبه اول به صورت ترکیب خطی از همه جواب های

[۱۰]

$$x_k(t) = e^{s_k t} = e^{\left(\frac{1}{h} W_k(-a_d h e^{ah}) - a\right)t}, k = 0, \pm 1, \pm 2, \dots \quad (۱۰-۳)$$

و به صورت،

$$x(t) = \sum_{k=-\infty}^{\infty} e^{\left(\frac{1}{h} W_k(-a_d h e^{ah}) - a\right)t} C_k^I \quad (۱۱-۳)$$

است. رابطه (۱۱-۳) را می توان به صورت زیر نیز نشان داد،

$$x(t) = \sum_{k=-\infty}^{\infty} x_k(t) C_k^I, t \geq -h \quad (۱۲-۳)$$

و از این که،

$$x(t) = g(t), t \in [-h, 0] \quad (۱۳-۳)$$

بدست می آوریم،

$$g(t) = \sum_{k=-\infty}^{\infty} x_k(t) C_k^I, t \in [-h, 0] \quad (۱۴-۳)$$

برای محاسبه ضرایب C_k^I ابتدا بازه $[-h, 0]$ را به صورت زیر به $2N$ قسمت، تقسیم می کنیم،

$$[-h, 0] = \left[-h, -h + \frac{h}{2N}\right] \cup \left[-h + \frac{h}{2N}, -h + \frac{2h}{2N}\right] \cup \dots \cup \left[\frac{-h}{2N}, 0\right]$$

و سپس با محاسبه مقدار تابع $g(t)$ در نقاط ابتدا و انتهایی بازه ها بدست می آوریم،

$$\begin{aligned}
g(0) &= \dots + C_{-N}^I x_{-N}(0) + C_{-(N-1)}^I x_{-(N-1)}(0) + \dots + C_{-1}^I x_{-1}(0) + C_0^I x_0(0) + C_1^I x_1(0) \\
&\quad + \dots + C_{(N-1)}^I x_{N-1}(0) + C_N^I x_N(0) + \dots, \\
g\left(\frac{-h}{2N}\right) &= \dots + C_{-N}^I x_{-N}\left(\frac{-h}{2N}\right) + C_{-(N-1)}^I x_{-(N-1)}\left(\frac{-h}{2N}\right) + \dots + C_{-1}^I x_{-1}\left(\frac{-h}{2N}\right) + C_0^I x_0\left(\frac{-h}{2N}\right) \\
&\quad + C_1^I x_1\left(\frac{-h}{2N}\right) + \dots + C_{(N-1)}^I x_{N-1}\left(\frac{-h}{2N}\right) + C_N^I x_N\left(\frac{-h}{2N}\right) + \dots, \\
g\left(\frac{-2h}{2N}\right) &= \dots + C_{-N}^I x_{-N}\left(\frac{-2h}{2N}\right) + C_{-(N-1)}^I x_{-(N-1)}\left(\frac{-2h}{2N}\right) + \dots + C_{-1}^I x_{-1}\left(\frac{-2h}{2N}\right) \\
&\quad + C_0^I x_0\left(\frac{-2h}{2N}\right) + C_1^I x_1\left(\frac{-2h}{2N}\right) + \dots + C_{(N-1)}^I x_{N-1}\left(\frac{-2h}{2N}\right) + C_N^I x_N\left(\frac{-2h}{2N}\right) + \dots
\end{aligned}$$

.....

$$\begin{aligned}
g(-h) &= \dots + C_{-N}^I x_{-N}(-h) + C_{-(N-1)}^I x_{-(N-1)}(-h) + \dots + C_{-1}^I x_{-1}(-h) + C_0^I x_0(-h) \\
&\quad + C_1^I x_1(-h) + \dots + C_{(N-1)}^I x_{N-1}(-h) + C_N^I x_N(-h) + \dots,
\end{aligned}$$

اکنون این معادلات را می توان به صورت ماتریسی،

$$\underbrace{\begin{pmatrix} g(0) \\ g\left(\frac{-h}{2N}\right) \\ g\left(\frac{-2h}{2N}\right) \\ \vdots \\ g(-h) \end{pmatrix}}_{G(h,N)} \approx \underbrace{\begin{pmatrix} x_{-N}(0) & \dots & x_N(0) \\ x_{-N}\left(\frac{-h}{2N}\right) & \dots & x_N\left(\frac{-h}{2N}\right) \\ x_{-N}\left(\frac{-2h}{2N}\right) & \dots & x_N\left(\frac{-2h}{2N}\right) \\ \vdots & \ddots & \vdots \\ x_{-N}(-h) & \dots & x_N(-h) \end{pmatrix}}_{X(h,N)} \underbrace{\begin{pmatrix} C_{-N}^I \\ C_{-(N-1)}^I \\ C_{-(N-2)}^I \\ \vdots \\ C_N^I \end{pmatrix}}_{C(N)} \quad (۱۵-۳)$$

یا به صورت،

$$G(h, N) \approx X(h, N)C(N) \quad (۱۶-۳)$$

نوشت که در آن بردار $C(N)$ تقریبی از بردار $(C_k^I)^T = (\dots, C_{-N}^I, \dots, C_{-1}^I, C_0^I, C_1^I, \dots, C_N^I, \dots)$ و برای مقادیرهای بزرگ از N می باشد (در اینجا T ، عملگر ترانهاده است).

با فرض وارون پذیری ماتریس $X(h, N)$ که یک ماتریس $(2N + 1) \times (2N + 1)$ می باشد، بدست می

آوریم،

$$C(N) \approx X^{-1}(h, N)G(h, N) \quad (17-3)$$

بنابراین ضرایب C_k^I را می توان از رابطه زیر بدست آورد،

$$C_k^I = \lim_{N \rightarrow \infty} \{X^{-1}(h, N)G(h, N)\}_k \quad (18-3)$$

همچنین می توان از تبدیل لاپلاس برای محاسبه ضرایب C_k^I استفاده کرد که در ادامه به توضیح آن می پردازیم.

ابتدا تبدیل لاپلاس از معادله سیستم (1-3) را به صورت زیر بدست می آوریم،

$$\begin{aligned} sX(s) - x_0 + a_d e^{-sh} X(s) + a_d \int_0^h e^{-st} g(t-h) dt + aX(s) \\ = (s + a_d e^{-sh} + a)X(s) - x_0 + a_d \int_0^h e^{-st} g(t-h) dt = 0 \end{aligned} \quad (19-3)$$

بنابراین،

$$X(s) = \frac{x_0 - a_d \int_0^h e^{-st} g(t-h) dt}{s + a_d e^{-sh} + a} \quad (20-3)$$

همچنین با گرفتن تبدیل لاپلاس از معادله (11-3) خواهیم داشت،

$$X(s) = \sum_{k=-\infty}^{\infty} \frac{C_k^I}{s - s_k} \quad (21-3)$$

$X(s)$ در معادله (21-3) را به صورت زیر در نظر می گیریم،

$$X(s) = \sum_{k=-\infty}^{\infty} \frac{C_k^I n_k(s)}{d(s)} \quad (22-3)$$

و چند جمله ای های $n_k(s)$ و $d(s)$ را به صورت زیر تعریف می کنیم،

$$d(s) = \prod_{k=-\infty}^{\infty} (s - s_k), \quad n_k(s) = \frac{d(s)}{(s - s_k)} \quad (23-3)$$

سپس با مقایسه صورت و مخرج کسرها در معادله های (20-3) و (22-3) نتیجه می گیریم،

$$d(s) = \prod_{k=-\infty}^{\infty} (s - s_k) = J(s)(s + a_d e^{-sh} + a) \quad (24-3)$$

9

$$\sum_{k=-\infty}^{\infty} C_k^l n_k(s) = J(s) \left(x_0 - a_d \int_0^h e^{-st} g(t-h) dt \right) \quad (25-3)$$

که در آن $J(s)$ چند جمله ای نامعین بر حسب s می باشد. با توجه به معادله (23-3)، $n_k(s)$ دارای ویژگی جالب و مفیدی به صورت زیر می باشد،

$$\begin{aligned} n_k(s = s_l) &= 0, & k \neq l & \quad (26-3) \\ &= \dots (s_l - s_{k-2})(s_l - s_{k-1})(s_l - s_{k+1})(s_l - s_{k+2}) \dots, & k = l \end{aligned}$$

با استفاده از این ویژگی و با قرار دادن $s = s_k$ در معادله (25-3) بدست می آوریم،

$$\begin{aligned} \dots, C_0^l &= \frac{J(s_0) \left(x_0 - a_d \int_0^h e^{-s_0 t} g(t-h) dt \right)}{n_0(s_0)} \\ &= \frac{J(s_0) \left(x_0 - a_d \int_0^h e^{-s_0 t} g(t-h) dt \right)}{\dots (s_0 - s_{-2})(s_0 - s_{-1})(s_0 - s_1) \dots}, \dots \end{aligned} \quad (27-3)$$

با توجه به اینکه داریم،

$$J(s) = \frac{\prod_{k=-\infty}^{\infty} (s - s_k)}{(s + a_d e^{-sh} + a)} \quad (28-3)$$

بنابراین،

$$\begin{aligned} J(s_0) &= \lim_{s \rightarrow s_0} \frac{\prod_{k=-\infty}^{\infty} (s - s_k)}{(s + a_d e^{-sh} + a)} = \lim_{s \rightarrow s_0} \frac{\frac{\partial}{\partial s} \prod_{k=-\infty}^{\infty} (s - s_k)}{\frac{\partial}{\partial s} (s + a_d e^{-sh} + a)} \\ &= \frac{\dots (s_0 - s_{-2})(s_0 - s_{-1})(s_0 - s_1) \dots}{1 - a_d h e^{-s_0 h}} \end{aligned} \quad (29-3)$$

حال اگر نتیجه بدست آمده را در معادله (27-3) قرار دهیم خواهیم داشت،

$$C_k^l = \frac{x_0 - a_d \int_0^h e^{-s_k t} g(t-h) dt}{1 - a_d h e^{-s_k h}} \quad (30-3)$$

به این ترتیب معادله برای بدست آوردن ضرایب C_k^l بدست می آید.

مثال ۱-۳ سیستم هیدرولیکی در مثال ۱-۲ را در نظر بگیرید. با فرض این که $A = 10m^2$ ،

$\rho = 1000kgm^{-3}$ ، $g = 10ms^{-2}$ ، $h = 1s$ ، $R = 10m^{\frac{-1}{2}}kg^{\frac{-1}{2}}$ و $k = -1000$ باشد سیستم حلقه

بسته به صورت زیر بدست می آید،

$$\dot{x}(t) = -\frac{g}{AR}x(t) + \frac{k}{A\rho}x(t-h) = -0.1x(t) - 0.1x(t-1)$$

با در نظر گرفتن شرایط اولیه به صورت،

$$x(t) = g(t) = 0, t \in [-1,0)$$

$$x(t) = 10, t = 0$$

به محاسبه s_k و ضرایب C_k^I می پردازیم. برای مثال، زمانی که $k = 0$ است با استفاده از نرم افزار

MATLAB بدست می آوریم،

$$s_0 = \frac{1}{h}W_0(a_d h e^{-ah}) + a = W_0(-0.1e^{0.1}) - 0.1 = -0.2253, C_0^I = 11.4321$$

نتایج برای برخی از شاخه های دیگر نیز با استفاده از این نرم افزار محاسبه می شوند که در جدول ۱-۳ نشان داده شده است.

جدول ۱-۳: نتایج بدست آمده برای محاسبه جواب در مثال ۱-۳ با استفاده از تابع لامبرت

| k | $k = -2$ | $k = -1$ | $k = 0$ | $k = 1$ | $k = 2$ |
|---------|----------------------|------------|-----------|----------------------|-----------------------|
| s_k | $-4.4439 - 7.3183 i$ | -3.53722 | -0.2253 | $-4.4439 + 7.3183 i$ | $-4.9861 + 13.7968 i$ |
| C_k^I | $-0.5165 + 1.1304 i$ | -4.1030 | 11.4321 | $-0.5165 - 1.1304 i$ | $-0.1891 - 0.6715 i$ |

به این ترتیب جواب سیستم از معادله،

$$x(t) = \sum_{k=-\infty}^{\infty} e^{s_k t} C_k^I$$

تعیین می شود که این جواب می تواند با استفاده از تعداد متناهی شاخه نیز تقریب زده شود.

۳-۱-۲ حالت ماتریسی

معادله دیفرانسیل تاخیری همگن زیر را در نظر می گیریم،

$$\begin{aligned} \dot{x}(t) + Ax(t) + A_d x(t-h) &= 0, h > 0, t > 0 \\ x(t) &= g(t), t \in [-h, 0] \end{aligned} \quad (۳۱-۳)$$

که در آن $x(t)$ یک بردار حالت $n \times 1$ است و A, A_d نیز ماتریس های $n \times n$ هستند.

فرض می کنیم جواب این معادله به صورت زیر باشد،

$$x(t) = e^{St} C^I \quad (۳۲-۳)$$

که در آن S یک ماتریس $n \times n$ با درایه های مختلط می باشد و C^I نیز یک بردار ثابت $n \times 1$ است.

با قرار دادن $x(t)$ در معادله سیستم بدست می آوریم،

$$(Se^{St} + Ae^{St} + A_d e^{S(t-h)})C^I = 0 \rightarrow (S + A + A_d e^{-Sh})e^{St}C^I = 0 \quad (۳۳-۳)$$

چون $e^{St}C^I \neq 0$ آنگاه نتیجه می گیریم،

$$(S + A + A_d e^{-Sh}) = 0 \quad (۳۴-۳)$$

برای حالت خاص که $A_d = 0$ باشد، سیستم (۳۱-۳) تبدیل به یک معادله دیفرانسیل معمولی می شود

و بنابراین معادله (۳۴-۳) به صورت زیر نوشته می شود،

$$S + A = 0 \leftrightarrow S = -A \quad (۳۵-۳)$$

با قرار دادن (۳۵-۳) در معادله (۳۲-۳) و با این فرض که $C^I = x_0$ باشد، نتیجه می گیریم که،

$$x(t) = e^{-At} x_0 \quad (۳۶-۳)$$

که همان فرم جواب سیستم همگن از معادلات دیفرانسیل معمولی می باشد.

در ادامه با ضرب رابطه (۳۴-۳) در e^{Sh} خواهیم داشت،

$$(S + A)e^{Sh} = -A_d \quad (۳۷-۳)$$

همچنین رابطه (۳۷-۳) را در he^{Ah} ضرب کرده و بدست می آوریم،

$$(S + A)he^{Sh}e^{Ah} = -A_dhe^{Ah} \quad (۳۸-۳)$$

در صورتی که ماتریسهای A, A_d جابجا شوند و یا به عبارت دیگر داشته باشیم $AA_d = A_dA$ ، آنگاه ماتریس های S, A نیز جابجا خواهند شد [۳۲] بنابراین معادله (۳۸-۳) به صورت،

$$(S + A)he^{Sh}e^{Ah} = (S + A)he^{(S+A)h} \quad (۳۹-۳)$$

نوشته می شود زیرا رابطه $e^Xe^Y = e^{X+Y}$ تنها زمانی برقرار است که ماتریس های X, Y با هم جابجا شوند (برای مثال $XY = YX$) بنابراین،

$$(S + A)he^{(S+A)h} = -A_dhe^{Ah} \quad (۴۰-۳)$$

طبق تعریف تابع لامبرت و از رابطه (۴۰-۳) نتیجه می گیریم،

$$(S + A)h = W(-A_dhe^{Ah}) \quad (۴۱-۳)$$

با حل معادله (۴۱-۳) برای S بدست می آوریم،

$$S = \frac{1}{h}W(-A_dhe^{Ah}) - A \quad (۴۲-۳)$$

بنابراین،

$$S_k = \frac{1}{h}W_k(-A_dhe^{Ah}) - A \quad (۴۳-۳)$$

و

$$x(t) = \sum_{k=-\infty}^{\infty} e^{\left(\frac{1}{h}W_k(-A_dhe^{Ah}) - A\right)t} \mathbf{C}_k^I \quad (۴۴-۳)$$

که این جواب مشابه فرم $x(t)$ برای حالت اسکالر است.

اما در حالت کلی ماتریس های A, A_d جابجا نمی شوند بنابراین S و A نیز جابجا نخواهند شد به همین دلیل،

$$(S + A)he^{Sh}e^{Ah} \neq (S + A)he^{(S+A)h} \quad (45-3)$$

به این ترتیب برای نوشتن جواب به صورت عبارت هایی از ماتریس تابع لامبرت که در رابطه،

$$W(H)e^{W(H)} = H \quad (46-3)$$

صدق می کند و در آن H یک ماتریس $n \times n$ می باشد، ماتریس مجهول Q را به گونه ای معرفی می کنیم که در رابطه زیر صدق کند،

$$(S + A)he^{(S+A)h} = -A_d h Q \quad (47-3)$$

با مقایسه رابطه (46-3) با (47-3) روابط زیر را بدست می آوریم،

$$(S + A)h = W(-A_d h Q) \quad (48-3)$$

و

$$W(-A_d h Q)e^{W(-A_d h Q)} = -A_d h Q \quad (49-3)$$

که در آن $W(-A_d h Q)$ مقدار تابع لامبرت را در $H = -A_d h Q$ نشان می دهد. با حل معادله (48-3) برای S بدست می آوریم،

$$S = \frac{1}{h} W(-A_d h Q) - A \quad (50-3)$$

همچنین با قرار دادن رابطه (50-3) در (34-3) خواهیم داشت،

$$W(-A_d h Q)e^{W(-A_d h Q) - Ah} = -A_d h \quad (51-3)$$

که از این رابطه برای محاسبه ماتریس Q استفاده می کنیم و سپس با قرار دادن آن در رابطه (50-3) به محاسبه ماتریس S می پردازیم.

به این ترتیب جواب کلی معادله دیفرانسیل تاخیری به صورت ترکیب خطی از،

$$x_k(t) = e^{S_k t} = e^{\frac{1}{h} W_k(-A_d h Q_k) - A t}, k = 0, \pm 1, \pm 2, \dots \quad (52-3)$$

و به صورت،

$$x(t) = \sum_{k=-\infty}^{\infty} e^{S_k t} C_k^I \quad (53-3)$$

بدست می آید. در اینجا بردار C_k^I یک بردار $n \times 1$ است و مشابه ضرایب C_k^I در حالت اسکالر محاسبه می شود. ماتریس S_k نیز در رابطه (53-3) به صورت،

$$S_k = \frac{1}{h} W_k(-A_d h Q_k) - A \quad (54-3)$$

می باشد که در آن مقدار ماتریس Q بدست آمده برای k امین شاخه از تابع لامبرت است که با استفاده از روش های عددی مانند تابع `fsolve` در نرم افزار MATLAB و از رابطه،

$$W_k(-A_d h Q_k) e^{W_k(-A_d h Q_k) - A h} = -A_d h \quad (55-3)$$

محاسبه می شود که W_k در این معادله، ماتریس تابع لامبرت را نشان می دهد به طوریکه هرگاه آرگومان تابع لامبرت یک ماتریس مختلط مربعی باشد، می توان ماتریس تابع لامبرت را معرفی کرد که در رابطه،

$$W_k(H_k) e^{W_k(H_k)} = H_k \quad (56-3)$$

صدق می کند.

حال قرار می دهیم $H_k = -A_d h Q_k$ و با به کار بردن تبدیلات تشابهی، H_k را به صورت زیر تبدیل می کنیم،

$$H_k = T_k J_k T_k^{-1} \quad (57-3)$$

که در آن فرم جردن J_k و H_k و T_k ماتریس مُدال می باشد. اگر مقدارهای ویژه λ_{ki} از ماتریس H_k متمایز باشند آنگاه ماتریس J_k یک ماتریس قطری با درایه های قطری λ_{ki} است و $W_k(H_k)$ به صورت زیر محاسبه می شود [18]،

$$W_k(H_k) = T_k W_k(J_k) T_k^{-1} = T_k \{diag(W_k(\lambda_{k1}), W_k(\lambda_{k2}), \dots, W_k(\lambda_{kn}))\} T_k^{-1} \quad (58-3)$$

و در صورتی که مقدارهای ویژه λ_{ki} از ماتریس J_k متمایز نباشند $W_k(H_k)$ از رابطه زیر بدست می آید،

$$W_k(H_k) = T_k \{diag(W_k(J_{k1}(\lambda_{k1})), W_k(J_{k2}(\lambda_{k2})), \dots, W_k(J_{ks}(\lambda_{ks})))\} T_k^{-1} \quad (59-3)$$

که در آن،

$$W_k(J_{ki}(\lambda_{ki})) = \begin{bmatrix} W_k(\lambda_{ki}) & W_k'(\lambda_{ki}) & \dots & \frac{1}{(m-1)!} W_k^{(m-1)}(\lambda_{ki}) \\ 0 & W_k(\lambda_{ki}) & \dots & \frac{1}{(m-2)!} W_k^{(m-2)}(\lambda_{ki}) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & W_k(\lambda_{ki}) \end{bmatrix} \quad (60-3)$$

و $J_{ki}(\lambda_{ki})$ یک بلوک جردن $m \times m$ است و m مرتبه تکرار مقدار ویژه λ_{ki} می باشد.

برای مثال $W_1\left(\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}\right), W_{-1}\left(\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 3 \end{bmatrix}\right)$ را محاسبه می کنیم.

برای $H = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$ ماتریس های مُدال و قطری شده به صورت زیر هستند،

$$T = \begin{bmatrix} -0.9094 & -0.5658 \\ 0.4160 & -0.8246 \end{bmatrix}, J = \begin{bmatrix} -0.3723 & 0 \\ 0 & 5.3723 \end{bmatrix}$$

به طوریکه ستونهای ماتریس T بردارهای ویژه نظیر مقادیر ویژه متمایزند. بنابراین بدست می آوریم،

$$\begin{aligned} W_1(J) &= W_1\left(\begin{bmatrix} -0.3723 & 0 \\ 0 & 5.3723 \end{bmatrix}\right) = \begin{bmatrix} W_1(-0.3723) & 0 \\ 0 & W_1(5.3723) \end{bmatrix} \\ &= \begin{bmatrix} -3.0765 + 7.4630i & 0 \\ 0 & 0.1251 + 4.7388i \end{bmatrix} \end{aligned}$$

در نتیجه $W_1(H)$ به صورت زیر محاسبه می شود،

$$W_1(H) = TW_1(J)T^{-1} = \begin{bmatrix} -2.3117 + 6.8122i & 1.6720 - 1.4227i \\ 1.1147 - 0.9484i & -0.6397 + 5.3896i \end{bmatrix}$$

برای $H = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 3 \end{bmatrix}$ نیز ماتریس های مُدال و قطری شده با استفاده از نرم افزار MATLAB به

صورت زیر بدست می آید،

$$T = \begin{bmatrix} 1 & -2 & -1 \\ 1 & 0 & -2 \\ 1 & 0 & 0 \end{bmatrix}, J = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

در نتیجه خواهیم داشت،

$$W_{-1}(J) = W_{-1} \left(\begin{bmatrix} 3 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \right) = \begin{bmatrix} W_{-1}(3) & 0 & 0 \\ 0 & W_{-1}(1) & W'_{-1}(1) \\ 0 & 0 & W_{-1}(1) \end{bmatrix}$$

$$= \begin{bmatrix} -0.4358 - 4.6183i & 0 & 0 \\ 0 & -1.5339 - 4.3752i & 1.0275i - 0.2235i \\ 0 & 0 & -1.5339 - 4.3752i \end{bmatrix}$$

بنابراین $W_{-1}(H)$ به صورت زیر بدست می آید،

$$W_{-1}(H) = TW_{-1}(J)T^{-1}$$

$$= \begin{bmatrix} -1.5339 - 4.3752i & 1.0257 - 0.2252i & 0.0706 - 0.0179i \\ 0 & -1.5339 - 4.3752i & 1.0981 - 0.2431i \\ 0 & 0 & -0.4358 - 4.6183i \end{bmatrix}$$

مثال ۲-۳ سیستم زیر را در نظر بگیرید،

$$\dot{x}(t) = \begin{bmatrix} -1 & -3 \\ 2 & -5 \end{bmatrix} x(t) + \begin{bmatrix} 1.66 & -0.697 \\ 0.93 & -0.330 \end{bmatrix} x(t-1)$$

ابتدا ماتریس های Q_k و سپس S_k را برای $k = -1, 0, 1$ با استفاده از نرم افزار MATLAB و از روابط

(۵۵-۳) و (۵۴-۳) محاسبه می کنیم. مقدارهای ویژه λ_{ki} از S_k نیز در جدول ۲-۳ نشان داده شده است.

جدول ۲-۳: نتایج بدست آمده برای محاسبه جواب در مثال ۲-۳ با استفاده از تابع لامبرت

| k | $k = 0$ | $k = \pm 1$ |
|----------------|--|---|
| Q_k | $\begin{bmatrix} -9.9183 & 14.2985 \\ -32.7746 & 6.5735 \end{bmatrix}$ | $\begin{bmatrix} -18.8024 \mp 10.2243i & 6.0782 \mp 2.2661i \\ -61.1342 \mp 23.681i & 1.0161 \mp 0.2653i \end{bmatrix}$ |
| S_k | $\begin{bmatrix} 0.3055 & -1.4150 \\ 2.1317 & -3.3015 \end{bmatrix}$ | $\begin{bmatrix} -0.3499 \pm 4.9801i & -1.6253 \mp 0.1459i \\ 2.4174 \mp 0.1308i & -5.1048 \pm 4.5592i \end{bmatrix}$ |
| λ_{ki} | $\begin{cases} -1.0119 \\ -1.9841 \end{cases}$ | $\begin{cases} -1.3990 \pm 5.0935i \\ -4.0558 \pm 4.4458i \end{cases}$ |

با استفاده از مقدارهای S_k در جدول ۲-۳ جواب سیستم از رابطه،

$$x(t) = \sum_{k=-\infty}^{\infty} e^{S_k t} C_k^I = \dots + e^{S_{-1}t} C_{-1}^I + e^{S_0 t} C_0^I + e^{S_1 t} C_1^I + \dots$$

بدست می آید که در آن ضرایب C_k^I را می توان با استفاده از نرم افزار MATLAB و شرایط اولیه،

$$x(t) = g(t), t \in [-h, 0)$$

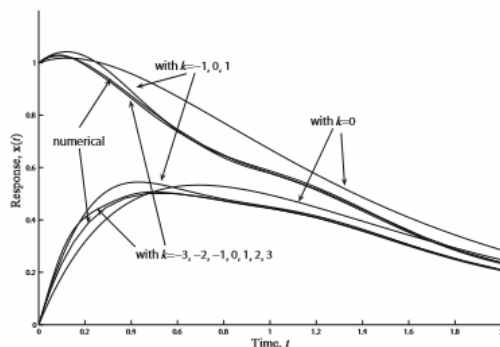
$$x(t) = x_0, t = 0$$

محاسبه کرد. فرض می کنیم برای $h = 1s, k = -1, 0, 1$ داشته باشیم $x_0 = g(t) = [1; 0]$ با استفاده

از این نرم افزار بدست می آوریم،

$$C_{-1}^I = [1.8003 + 1.9750i, 5.6819 + 4.1444i], C_0^I = [-0.0327, -2.7724], C_1^I = [1.8003 - 1.9750i, 5.6819 - 4.1444i]$$

در شکل ۱-۳ نتایج بدست آمده با جوابی که برای این سیستم و با استفاده از انتگرال گیری عددی بدست می آید، مقایسه شده است.



شکل ۱-۳: مقایسه جواب بدست آمده از انتگرال گیری عددی و تابع لامبرت [۳۲]

مشاهده می شود هر چه تعداد شاخه ها بیشتر می شود جوابی که با استفاده از تابع لامبرت بدست می آید با جواب بدست آمده از انتگرال گیری عددی، مطابقت بیشتری دارد.

۲-۳ سیستم های ناهمگن

۲-۳-۱ حالت اسکالر

معادله دیفرانسیل تاخیری زیر را در نظر می گیریم،

$$\dot{x}(t) + ax(t) + a_ax(t-h) = bu(t), t > 0, h > 0 \quad (۶۱-۳)$$

$$x(t) = g(t), t \in [-h, 0]$$

جواب این معادله ناهمگن در [۱۴] بدست آمده است که به صورت،

$$x(t) = \int_0^t \psi(t, \xi) bu(\xi) d\xi \quad (۶۲-۳)$$

می باشد که در آن تابع $\psi(t, \xi)$ باید در شرایط زیر صدق کند،

$$a) \frac{\partial}{\partial \xi} \psi(t, \xi) = \begin{cases} -a\psi(t, \xi), & t-h \leq \xi < t \\ -a\psi(t, \xi) - a_a\psi(t, \xi+h), & \xi < t-h \end{cases} \quad (۶۳-۳)$$

$$b) \psi(t, t) = 1$$

$$c) \psi(t, \xi) = 0, \xi > t$$

تابع اولیه $\psi(t, \xi)$ را می توان براساس روش تابع لامبرت محاسبه کرد. یک تابع $\psi(t, \xi)$ که در شرط اول صدق می کند به صورت زیر است،

$$\psi(t, \xi) = e^{a(t-\xi)} \quad (64-3)$$

تابع دیگری که در شرط دوم صدق می کند به صورت،

$$\psi(t, \xi)_k = e^{\left(\frac{1}{h} W_k(-a_d h e^{ah}) - a\right)(t-\xi)} = e^{s_k(t-\xi)}, k = -\infty, \dots, \infty \quad (65-3)$$

می باشد. چون تعداد نامتناهی جواب برای شاخه های تابع لامبرت وجود دارد بنابراین جواب کلی می تواند به صورت مجموع زیر نوشته شود،

$$\psi(t, \xi) = \sum_{k=-\infty}^{\infty} e^{s_k(t-\xi)} C_k^N \quad (66-3)$$

به این ترتیب داریم،

$$a) \psi(t, \xi) = \begin{cases} e^{a(t-\xi)}, & t-h \leq \xi < t \\ \sum_{k=-\infty}^{\infty} e^{s_k(t-\xi)} C_k^N, & \xi < t-h \end{cases} \quad (67-3)$$

$$b) \psi(t, \xi) = 0, \xi > t$$

در نتیجه جواب معادله ناهمگن به صورت زیر نوشته می شود،

حالت اول: $0 \leq t \leq h$

$$x(t) = \int_0^t e^{a(t-\xi)} bu(\xi) d\xi \quad (68-3)$$

حالت دوم: $t > h$

$$x(t) = \int_0^{t-h} \sum_{k=-\infty}^{\infty} e^{s_k(t-\xi)} C_k^N bu(\xi) d\xi + \int_{t-h}^t e^{a(t-\xi)} bu(\xi) d\xi \quad (69-3)$$

حال با تقسیم بازه $[0, h]$ به $2N$ قسمت و همچنین با استفاده از روابط (68-3) و (69-3) به محاسبه C_k^N

می پردازیم،

$$\underbrace{\begin{pmatrix} \sigma(h) \\ \sigma\left(h - \frac{h}{2N}\right) \\ \sigma\left(h - \frac{2h}{2N}\right) \\ \vdots \\ \sigma(0) \end{pmatrix}}_{\bar{\sigma}} = \underbrace{\begin{pmatrix} \eta_{-N}(h) & \cdots & \eta_N(h) \\ \eta_{-N}\left(h - \frac{h}{2N}\right) & \cdots & \eta_N\left(h - \frac{h}{2N}\right) \\ \eta_{-N}\left(h - \frac{2h}{2N}\right) & \cdots & \eta_N\left(h - \frac{2h}{2N}\right) \\ \vdots & \ddots & \vdots \\ \eta_{-N}(0) & \cdots & \eta_N(0) \end{pmatrix}}_{\bar{\eta}} \begin{pmatrix} C_{-N}^N \\ C_{-(N-1)}^N \\ C_{-(N-2)}^N \\ \vdots \\ C_N^N \end{pmatrix} + \underbrace{\begin{pmatrix} \delta(h) \\ \delta\left(h - \frac{h}{2N}\right) \\ \delta\left(h - \frac{2h}{2N}\right) \\ \vdots \\ \delta(0) \end{pmatrix}}_{\bar{\delta}} \quad (70-3)$$

که در آن،

$$\sigma(t) = \int_0^t e^{a(t-\xi)} bu(\xi) d\xi$$

$$\eta(t) = \int_0^{t-h} \sum_{k=-\infty}^{\infty} e^{s_k(t-\xi)} C_k^N bu(\xi) d\xi \quad (71-3)$$

$$\delta(t) = \int_{t-h}^t e^{a(t-\xi)} bu(\xi) d\xi$$

به این ترتیب C_k^N به صورت زیر نشان داده می شود،

$$C_k^N = \lim_{N \rightarrow \infty} [\bar{\eta}^{-1}(h, N) \cdot (\bar{\sigma} - \bar{\delta})]_k \quad (72-3)$$

همچنین مشابه سیستم های همگن برای حالت اسکالر و با استفاده از تبدیل لاپلاس می توان نشان

داد که ضرایب C_k^N را از رابطه،

$$C_k^N = \frac{1}{1 - a_d h e^{-s_k h}} \quad (73-3)$$

نیز می توان بدست آورد. حال با استفاده از روابط بالا و ضریب C_k^N بدست آمده جواب معادله ناهمگن

را برای $t \in [0, h]$ به صورت زیر بدست می آوریم،

$$\int_0^t e^{a(t-\xi)} bu(\xi) d\xi = \int_0^{t-h} \sum_{k=-\infty}^{\infty} e^{s_k(t-\xi)} C_k^N bu(\xi) d\xi + \int_{t-h}^t e^{a(t-\xi)} bu(\xi) d\xi \quad (74-3)$$

$$\rightarrow \int_0^t e^{a(t-\xi)} bu(\xi) d\xi - \int_{t-h}^t e^{a(t-\xi)} bu(\xi) d\xi = \int_0^{t-h} \sum_{k=-\infty}^{\infty} e^{s_k(t-\xi)} C_k^N bu(\xi) d\xi$$

$$\rightarrow \int_0^{t-h} e^{a(t-\xi)} bu(\xi) d\xi = \int_0^{t-h} \sum_{k=-\infty}^{\infty} e^{s_k(t-\xi)} C_k^N bu(\xi) d\xi \quad (75-3)$$

$$\rightarrow \int_0^{t-h} e^{a(t-\xi)} bu(\xi) d\xi - \int_0^{t-h} \sum_{k=-\infty}^{\infty} e^{s_k(t-\xi)} C_k^N bu(\xi) d\xi = 0$$

$$\rightarrow \int_0^{t-h} \left\{ e^{a(t-\xi)} - \sum_{k=-\infty}^{\infty} e^{s_k(t-\xi)} C_k^N \right\} bu(\xi) d\xi = 0, t \in [0, h]$$

قرار می دهیم $\theta = t - \xi$ و بدست می آوریم،

$$\int_t^h \left\{ e^{a\theta} - \sum_{k=-\infty}^{\infty} e^{s_k\theta} C_k^N \right\} bu(t - \theta) d\theta = 0 \quad (76-3)$$

بنابراین نتیجه می گیریم،

$$e^{a\theta} = \sum_{k=-\infty}^{\infty} e^{s_k\theta} C_k^N, \theta \in [0, h]$$

و از این که $0 \leq \theta \leq h$ بدست می آوریم $0 \leq t - \xi \leq h$ در نتیجه $t - h \leq \xi \leq t$ بنابراین خواهیم داشت،

$$e^{a(t-\xi)} = \sum_{k=-\infty}^{\infty} e^{s_k(t-\xi)} C_k^N, \xi \in [t-h, t] \quad (77-3)$$

به این ترتیب از روابط (74-3) و (77-3) جواب معادله ناهمگن به صورت زیر حاصل می شود،

$$x(t) = \int_0^t \sum_{k=-\infty}^{\infty} e^{s_k(t-\xi)} C_k^N bu(\xi) d\xi \quad (78-3)$$

بنابراین جواب کلی معادله (61-3) به صورت زیر بدست می آید،

$$x(t) = \sum_{k=-\infty}^{\infty} e^{s_k t} C_k^I + \int_0^t \sum_{k=-\infty}^{\infty} e^{s_k(t-\xi)} C_k^N bu(\xi) d\xi \quad (79-3)$$

که در آن،

$$s_k = \frac{1}{h} W_k(-a_d h e^{ah}) - a \quad (۸۰-۳)$$

است.

مثال ۳-۳ سیستم زیر را در نظر بگیرید،

$$\dot{x}(t) = ax(t) + a_d x(t-h) + bu(t), t > 0$$

با فرض این که $a = a_d = -1$ و $h = 1s$ باشد و داشته باشیم،

$$bu(t) = \cos(t), t > 0$$

برای شرایط اولیه،

$$x(t) = g(t) = 1, t \in [-1, 0)$$

$$x(t) = x_0 = 1, t = 0$$

به محاسبه جواب سیستم می پردازیم.

ابتدا برای بدست آوردن جواب سیستم به محاسبه S_k, C_k^I, C_k^N می پردازیم. نتایج برای شاخه های

$k = -1, 0, 1$ که با استفاده از نرم افزار MATLAB بدست آمده اند، در جدول ۳-۳ نشان داده شده

است.

جدول ۳-۳: نتایج بدست آمده برای محاسبه جواب در مثال ۳-۳ با استفاده از تابع لامبرت

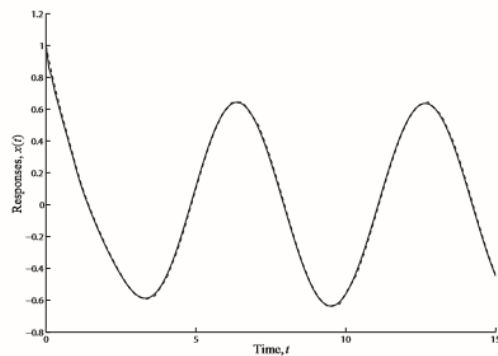
| k | ... | $k = -1$ | $k = 0$ | $k = 1$ | ... |
|---------|-----|---------------------|---------------------|---------------------|-----|
| s_k | ... | $-0.6050 - 1.7882i$ | $-0.6050 + 1.7882i$ | $-2.0528 + 7.7184i$ | ... |
| C_k^I | ... | $1.1014 + 0.8494i$ | $1.1014 - 0.8494i$ | $0.9669 - 0.2506i$ | ... |
| C_k^N | ... | $0.2712 + 0.3477i$ | $0.2712 - 0.3477i$ | $-0.0009 - 0.1296i$ | ... |

برای شاخه های دیگر نیز نتایج به همین ترتیب و با استفاده از نرم افزار MATLAB محاسبه می شوند.

بنابراین جواب سیستم از معادله زیر،

$$x(t) = \sum_{k=-\infty}^{\infty} e^{s_k t} C_k^I + \int_0^t \sum_{k=-\infty}^{\infty} e^{s_k(t-\xi)} C_k^N bu(\xi) d\xi$$

و برای تعداد متناهی شاخه تقریب زده می شود. شکل ۲-۳ نیز پاسخ بدست آمده برای سیستم را با روش جدید و برای هفت شاخه از تابع لامبرت (توپر) در مقایسه با جواب بدست آمده از انتگرال گیری عددی (خط تیره) نشان می دهد.



شکل ۲-۳: مقایسه جواب بدست آمده از انتگرال گیری عددی و تابع لامبرت [۳۲]

۲-۲-۳ حالت ماتریسی

معادله دیفرانسیل تاخیری ناهمگن زیر را در نظر بگیرید،

$$\dot{x}(t) + Ax(t) + A_d x(t-h) = Bu(t), t > 0, h > 0 \quad (۸۱-۳)$$

$$x(t) = g(t), t \in [-h, 0]$$

که در آن A, A_d ماتریس های $n \times n$ ، B یک ماتریس $n \times r$ ، $x(t)$ یک بردار حالت $n \times 1$ و $u(t)$ یک بردار ورودی $r \times 1$ است. جواب این معادله، مشابه حالت اسکالر و به صورت،

$$x(t) = \sum_{k=-\infty}^{\infty} e^{S_k t} C_k^I + \int_0^t \sum_{k=-\infty}^{\infty} e^{S_k(t-\xi)} C_k^N B u(\xi) d\xi \quad (۸۲-۳)$$

بدست می آید که در آن،

$$S_k = \frac{1}{h} W_k(-A_d h Q_k) - A \quad (۸۳-۳)$$

می باشد و C_k^I نیز یک بردار $n \times 1$ و وابسته به A, A_d, h و شرایط اولیه می باشد در حالیکه C_k^N یک ماتریس $n \times n$ و وابسته به A, A_d, h است اما به شرایط اولیه بستگی ندارد. هر دو این ضرایب نیز

مشابه حالت اسکالر محاسبه می شوند. ماتریس Q_k نیز در این معادله متناظر با هر شاخه از تابع لامبرت به روش عددی و با استفاده از تابع `fsolve` در نرم افزار MATLAB و برای شرایط اولیه متنوعی از معادله زیر بدست می آید،

$$W_k(-A_d h Q_k) e^{W_k(-A_d h Q_k) - A h} = -A_d h \quad (۸۴-۳)$$

مثال ۳-۴ سیستم زیر را در نظر بگیرید،

$$\dot{x}(t) = Ax(t) + A_d x(t-h) + Bu(t), t > 0, h > 0$$

با شرایط اولیه،

$$x(t) = g(t) = [1 \ 0]^T, t \in [-h, 0)$$

$$x(t) = x_0 = [1 \ 0]^T, t = 0$$

و با در نظر گرفتن $h = 1s$ و

$$A = \begin{bmatrix} -1 & -3 \\ 2 & -5 \end{bmatrix}, A_d = \begin{bmatrix} 1.66 & -0.697 \\ 0.93 & -0.330 \end{bmatrix}, Bu(t) = \begin{bmatrix} \cos(t) \\ 0 \end{bmatrix}, t > 0$$

ابتدا برای بدست آوردن جواب سیستم به محاسبه S_k, C_k^I, C_k^N می پردازیم که نتایج برای شاخه های $k = -1, 0, 1$ که با استفاده از نرم افزار MATLAB بدست آمده اند در جدول ۳-۴ نشان داده شده است.

جدول ۳-۴: نتایج بدست آمده برای محاسبه جواب در مثال ۳-۴ با استفاده از تابع لامبرت

| k | $k = 0$ | $k = \pm 1$ |
|---------|--|---|
| S_k | $\begin{bmatrix} 0.3055 & -1.4150 \\ 2.1317 & -3.3015 \end{bmatrix}$ | $\begin{bmatrix} -0.3499 \pm 4.9801i & -1.6253 \mp 0.1459i \\ 2.4174 \mp 0.1308i & -5.1048 \pm 4.5592i \end{bmatrix}$ |
| C_k^I | $\begin{bmatrix} -0.0327 \\ -2.7724 \end{bmatrix}$ | $\begin{bmatrix} 1.8003 \mp 1.9750i \\ 5.6819 \mp 4.1444i \end{bmatrix}$ |
| C_k^N | $\begin{bmatrix} 0.3424 & -0.0789 \\ -0.0563 & 0.4855 \end{bmatrix}$ | $\begin{bmatrix} 0.0767 \mp 0.1876i & 0.0142 \pm 0.419i \\ 0.0239 \mp 0.0013i & 0.0196 \mp 0.1925i \end{bmatrix}$ |

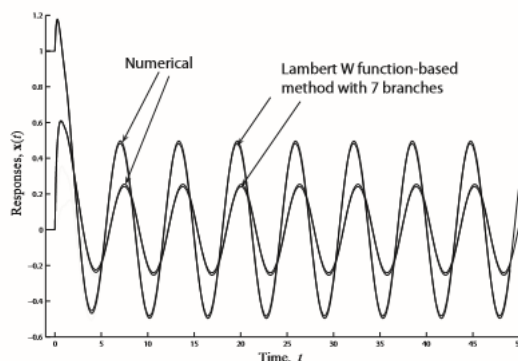
با به کار بردن این مقادیر در رابطه،

$$x(t) = \sum_{k=-\infty}^{\infty} e^{S_k t} C_k^I + \int_0^t \sum_{k=-\infty}^{\infty} e^{S_k(t-\xi)} C_k^N Bu(\xi) d\xi$$

جواب سیستم را بدست می آوریم. نتایج بدست آمده برای هفت شاخه از تابع لامبرت

۳-۳ در مقایسه با جواب بدست آمده از انتگرال گیری عددی در شکل ۳-۳

نشان داده شده است.



شکل ۳-۳: مقایسه جواب بدست آمده از انتگرال گیری عددی و تابع لامبرت [۳۲]

جدول ۵-۳ نیز مقایسه جواب معادلات دیفرانسیل معمولی و معادلات دیفرانسیل تاخیری را نشان می

دهد. همانطور که مشاهده می شود جوابی که برای معادلات دیفرانسیل تاخیری با استفاده از تابع

لامبرت بدست می آید شباهت ظاهری با جواب معادلات دیفرانسیل معمولی دارد.

جدول ۵-۳: مقایسه جواب معادلات دیفرانسیل معمولی و معادلات دیفرانسیل تاخیری

| ODEs | DDEs |
|--|---|
| <i>Scalar Case</i> | |
| $\dot{x}(t) = ax(t) + bu(t), t > 0$ | $\dot{x}(t) = ax(t) + a_d x(t-h) + bu(t), t > 0$ |
| $x(t) = x_0, t = 0$ | $x(t) = g(t), t \in [-h, 0); x(t) = x_0, t = 0$ |
| $x(t) = e^{at} x_0 + \int_0^t e^{a(t-\xi)} bu(\xi) d\xi$ | $x(t) = \sum_{k=-\infty}^{\infty} e^{s_k t} C_k^I + \int_0^t \sum_{k=-\infty}^{\infty} e^{s_k(t-\xi)} C_k^N bu(\xi) d\xi$ |
| | $s_k = \frac{1}{h} W_k(a_d h e^{-ah}) + a$ |
| <i>Matrix – Vector Case</i> | |
| $\dot{x}(t) = Ax(t) + Bu(t), t > 0$ | $\dot{x}(t) = Ax(t) + A_d x(t-h) + Bu(t), t > 0$ |
| $x(t) = x_0, t = 0$ | $x(t) = g(t), t \in [-h, 0); x(t) = x_0, t = 0$ |
| $x(t) = e^{At} x_0 + \int_0^t e^{A(t-\xi)} Bu(\xi) d\xi$ | $x(t) = \sum_{k=-\infty}^{\infty} e^{s_k t} C_k^I + \int_0^t \sum_{k=-\infty}^{\infty} e^{s_k(t-\xi)} C_k^N Bu(\xi) d\xi$ |
| | $S_k = \frac{1}{h} W_k(A_d h Q_k) + A$ |

فصل چهارم

بررسی پایداری سیستم های تاخیری به وسیله تابع لامبرت

در این فصل، ابتدا تعاریفی برای بررسی مفهوم پایداری بیان می کنیم سپس با استفاده از تابع لامبرت به بررسی پایداری سیستم های تاخیری در هر دو حالت اسکالر و ماتریسی می پردازیم.

۱-۴ پایداری

به طور کلی در بررسی پایداری یک سیستم می توان از تعاریف زیر استفاده کرد،

تعریف ۱-۴ یک سیستم تاخیری خطی، پایدار مجانبی است هر گاه،

$$\lim_{t \rightarrow \infty} x(t) = 0 \quad (1-4)$$

اگر بخواهیم پایداری سیستم تاخیری را با استفاده از معادله مشخصه آن بررسی کنیم از تعریف زیر استفاده می کنیم.

تعریف ۲-۴ یک سیستم تاخیری خطی، پایدار مجانبی است اگر همه جواب های معادله مشخصه سیستم در سمت چپ صفحه مختلط قرار گیرند.

۱-۱-۴ حالت اسکالر

یکی از فواید روش تابع لامبرت این است که موقعیت راست ترین مقدارهای ویژه را می توان از شاخه اصلی تابع لامبرت تعیین کرد و سپس از نتیجه بدست آمده برای تعیین پایداری یک سیستم از معادلات دیفرانسیل تاخیری استفاده کرد. برای حالت اسکالر ثابت می شود راست ترین قطب ها (مقدارهای ویژه) با استفاده از شاخه اصلی تابع لامبرت ($k = 0$) بدست می آیند، به عبارت دیگر داریم،

$$Re\{s_0\} = Re\left\{\frac{1}{h}W_0(a_d h e^{-ah}) + a\right\} \geq Re\{s_k\}, \forall k \neq 0 \quad (2-4)$$

بنابراین می توان از شاخه اصلی تنها برای تعیین پایداری استفاده کرد.

سیستم تاخیری اسکالر خطی زیر را در نظر بگیرید،

$$\dot{x}(t) = ax(t) + a_d x(t-h) \quad (3-4)$$

که در آن $a, a_d, x(t) \in \mathbb{C}$ و $h > 0$ می باشد. از تعریف ۲-۴ سیستم (۳-۴) پایدار مجانبی است اگر و تنها اگر ریشه های معادله مشخصه $s - a - a_d e^{-sh}$ در سمت چپ صفحه مختلط قرار گیرند.

برای بدست آوردن ریشه های معادله مشخصه با استفاده از تابع لامبرت به صورت زیر عمل می کنیم،

$$\begin{aligned} s - a - a_d e^{-sh} = 0 &\leftrightarrow s - a = a_d e^{-sh} \leftrightarrow (s - a) h e^{sh} e^{-ah} = a_d h e^{-ah} \\ &\leftrightarrow (s - a) h e^{(s-a)h} = a_d h e^{-ah} \end{aligned}$$

با استفاده از رابطه،

$$W(H)e^{W(H)} = H$$

بدست می آوریم،

$$(s - a)h = W(a_d h e^{-ah}) \leftrightarrow s = \frac{1}{h} W(a_d h e^{-ah}) + a$$

بنابراین خواهیم داشت،

$$s_k = \frac{1}{h} W_k(a_d h e^{-ah}) + a$$

حال به بیان یک لم می پردازیم.

لم ۴-۱ سیستم تاخیری اسکالر خطی، پایدار مجانبی است اگر و تنها اگر،

$$Re\left\{\frac{1}{h}W_0(a_d h e^{-ah}) + a\right\} < 0 \quad (4-4)$$

اثبات: بنا به خاصیت تابع لامبرت در معادله (۲-۷)، $\frac{1}{h}W_0(a_d h e^{-ah}) + a$ در میان همه ریشه های

معادله مشخصه یعنی $\frac{1}{h}W_k(a_d h e^{-ah}) + a, k = 0, \pm 1, \pm 2, \dots, \pm \infty$ راست ترین مقدار ویژه است.

بنابراین از تعریف ۲-۴ حکم بدست می آید [۲۱].

مثال ۱-۴ سیستم زیر را با $a = -1, a_d = 0.5$ و $h = 1s$ در نظر بگیرید،

$$\dot{x}(t) = ax(t) + a_d x(t-h)$$

معادله مشخصه این سیستم به صورت،

$$s + 1 - 0.5e^{-s} = 0$$

می باشد. با حل این معادله با استفاده از تابع لامبرت بدست می آوریم،

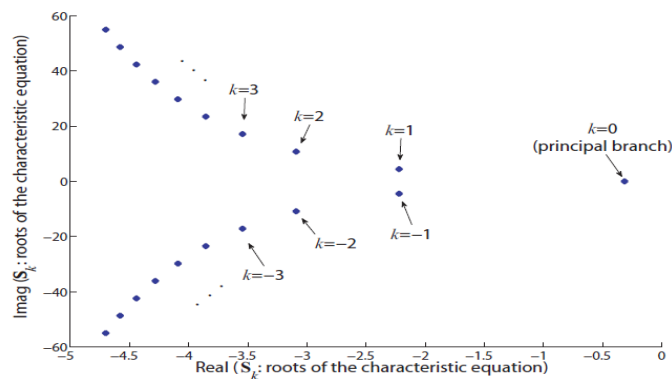
$$s_k = W_k(0.5e) - 1$$

به این ترتیب با استفاده از نرم افزار MATLAB برای شاخه های $k = -3, \dots, 3$ مقادیر ویژه محاسبه می شوند. نتایج نیز در جدول ۱-۴ نشان داده شده است.

جدول ۱-۴: مقادیر ویژه محاسبه شده برای سیستم در مثال ۱-۴ با استفاده از تابع لامبرت

| | |
|----------------------------|-------------------------------|
| $s_0 = -0.3149$ | |
| $s_1 = -2.2211 + 4.442i$ | $s_{-1} = -2.2211 - 4.442i$ |
| $s_2 = -3.0915 + 10.3044i$ | $s_{-2} = -3.0915 - 10.3044i$ |
| $s_3 = -3.5450 + 17.1313i$ | $s_{-3} = -3.5450 - 17.1313i$ |

مقادیر ویژه دیگر نیز به همین ترتیب قابل محاسبه هستند. زیرا همانطور که بیان کردیم به دلیل وجود عبارت نمایی در معادله مشخصه سیستم، تعداد مقادیر ویژه، نامتناهی است. شکل ۱-۴ نیز طیف ویژه این سیستم را نشان می دهد.



شکل ۱-۴: طیف ویژه سیستم (۳-۴) با $a = -1, a_d = 0.5$ و $h = 1s$ [۳۲]

واضح است که این سیستم، پایدار می باشد و راست ترین مقدار ویژه در سمت چپ صفحه مختلط (که اصطلاحاً نیم صفحه چپ (LHP)^۱ نامیده می شود) قرار گرفته است.

همانطور که مشاهده می شود روش لامبرت ابزاری برای تحلیل و کنترل سیستم های تاخیری فراهم می کند به گونه ای که هر مقدار ویژه که به صورت یک عبارت بر حسب پارامترهای a, a_d, h نشان داده می شود، متناظر با یک شاخه از تابع لامبرت است.

۴-۱-۲ حالت ماتریسی

سیستم تاخیری زیر را در نظر بگیرید،

$$\begin{aligned} \dot{x}(t) &= Ax(t) + A_d x(t-h), t \geq 0, h > 0 \\ x(t) &= g(t), t \in [-h, 0] \end{aligned} \quad (۵-۴)$$

فرم جواب این سیستم نشان می دهد که شرط پایداری آن به مقدارهای ویژه ماتریس S_k و همچنین ماتریس e^{S_k} وابسته است. به عبارت دیگر سیستم تاخیری، پایدار مجانبی است اگر و تنها اگر مقدارهای ویژه ماتریس S_k در سمت چپ صفحه مختلط قرار گیرند و یا به طور معادل در مفهوم لیپانوف همه ی مقدارهای ویژه ماتریس e^{S_k} درون دایره واحد قرار گیرند [۲۷]. از طرفی محاسبه ماتریس های S_k و یا e^{S_k} برای تعداد نامتناهی شاخه شدنی نیست در حالیکه تعیین راست ترین مقدارهای ویژه برای تشخیص پایداری در میان تعداد نامتناهی مقدار ویژه، اهمیت بسیاری دارد. همانطور که اشاره کردیم برای حالت اسکالر، ثابت شده است راست ترین قطب ها با استفاده از شاخه اصلی تابع لامبرت بدست می آیند. اثبات برای حالت اسکالر همچنین می تواند برای حالت ماتریسی، زمانی که A, A_d همزمان مثلثی شدنی^۲ باشند و یا در حالت خاص که داشته باشیم $AA_d = A_d A$ تعمیم داده شود [۱۹]. اگر چه تاکنون اثبات برای حالت کلی وجود ندارد اما در همه مثال های حل شده، مشاهده شده است که راست ترین مقدارهای ویژه با استفاده از m شاخه اول که در آن m پوچی

^۱Left Half Plane

^۲Simultaneously Triangularizable

A_d می باشد، بدست می آید [۶]. همچنین اگر ماتریس ضرایب A_d مقدارهای ویژه چندگانه صفر را نداشته باشد مشاهده شده است که راست ترین مقدارهای ویژه با استفاده از شاخه اصلی تابع لامبرت بدست می آیند و چون اثبات کاملی وجود ندارد معمولاً این مشاهدات به صورت یک حدس مانند زیر مطرح می شود که از آن علاوه بر تعیین پایداری سیستم هایی از معادلات دیفرانسیل تاخیری بلکه همچنین برای قرار دادن یک زیر مجموعه از مقدارهای ویژه در موقعیت های مورد نظر استفاده می کنیم.

حدس: اگر ماتریس ضرایب A_d مقدارهای ویژه چندگانه صفر را نداشته باشد آنگاه [۲۸]،

$$\max[\operatorname{Re}\{eig(S_0)\}] \geq \operatorname{Re}\{eig(S_k)\}, k = -\infty, \dots, 0, \dots, \infty \quad (۶-۴)$$

همچنین با بیان این که [۶]،

$$\max[\operatorname{Re}\{eig(S_{-m})\}, \dots, \operatorname{Re}\{eig(S_m)\}] \geq \max[\operatorname{Re}\{eig(S_k)\}], \forall k \quad (۷-۴)$$

که در آن $m = \operatorname{nullity}(A_d)$ می باشد، می توان حدس مطرح شده را به این صورت بیان کرد که اگر پوچی A_d صفر باشد آنگاه،

$$\max[\operatorname{Re}\{eig(S_0)\}] \geq \max[\operatorname{Re}\{eig(S_k)\}], \forall k \quad (۸-۴)$$

به عبارت دیگر اگر A_d مقدارهای ویژه چندگانه صفر را نداشته باشد و یا A_d رتبه کامل^۱ باشد می توان تنها از شاخه اصلی برای تعیین پایداری استفاده کرد در غیر این صورت همه m شاخه اول برای رسیدن به یک نتیجه برای پایداری سیستم باید بررسی شوند.

در صورتی که صفر یک مقدار ویژه چندگانه برای ماتریس A_d باشد، مشاهده شده است راست ترین مقدارهای ویژه با استفاده از شاخه اصلی ($k = 0$) و یا $k = \pm 1$ بدست می آیند.

همچنین برای رسیدن به یک شرط پایداری برای سیستم (۵-۴) احتیاج به فرض زیر داریم،

^۱Full Rank

فرض ۱-۴ چند جمله ای مشخصه سیستم تاخیری خطی (۵-۴) به صورت،

$$\det(sI - A - A_d e^{-sh}) = \prod_{i=1}^n (s - \alpha_i - \beta_i e^{-sh}) = 0 \quad (۹-۴)$$

است که در آن،

$$\alpha_i, \beta_i \in \mathbb{C}, i = 1, \dots, n$$

می باشند [۲۱].

تذکر ۱-۴ اگر A, A_d همزمان مثلثی شدند باشند که در مورد آن شرط لازم و کافی آن است که $AA_d - A_dA$ پوچتوان باشد آنگاه ماتریس نامنفردی وجود دارد که A, A_d را همزمان به ماتریس های مثلثی تبدیل می کند که در آن،

$$\alpha_i, \beta_i \in \mathbb{C}, i = 1, \dots, n$$

عناصر روی قطر ماتریس های مثلثی شده هستند [۲۱].

بنابراین در حالت خاص، زمانی که دو ماتریس A, A_d با هم جابجا می شوند یعنی داشته باشیم $AA_d = A_dA$ می توان گفت دو ماتریس همزمان مثلثی شدند زیرا ماتریس صفر، یک ماتریس پوچتوان است.

همچنین از این که $\prod_{i=1}^n (s - \alpha_i - \beta_i e^{-sh}) = 0$ واضح است که برای هر i داریم،

$$(s - \alpha_i - \beta_i e^{-sh}) = 0 \rightarrow s - \alpha_i = \beta_i e^{-sh} \rightarrow (s - \alpha_i) h e^{(s - \alpha_i) h} = \beta_i h e^{-\alpha_i h}$$

$$\rightarrow (s - \alpha_i) h = W(\beta_i h e^{-\alpha_i h}) \rightarrow s = \frac{1}{h} W(\beta_i h e^{-\alpha_i h}) + \alpha_i \rightarrow s_k = \frac{1}{h} W_k(\beta_i h e^{-\alpha_i h}) + \alpha_i$$

بنابراین اگر فرض ۱-۴ برقرار باشد از لم زیر یک شرط برای پایداری سیستم (۵-۴) به صورت زیر بدست می آید.

لم ۱-۴ بنا به فرض ۱-۴ سیستم تاخیری خطی، پایدار مجانبی است اگر و تنها اگر،

$$\operatorname{Re}\left\{\frac{1}{h}W_0(\beta_i h e^{-\alpha_i h}) + \alpha_i\right\} < 0 \quad (10-4)$$

باشد [۲۱].

همچنین الگوریتم زیر روش بررسی پایداری سیستم تاخیری (۵-۴) را در حالت کلی شرح می دهد.

الگوریتم ۱-۴

گام ۱- معادله (۱۱-۴) را برای شاخه اصلی ($k = 0$) و برای پیدا کردن Q_0 حل می کنیم،

$$W_k(-A_d h Q_k) e^{W_k(-A_d h Q_k) - Ah} = -A_d h \quad (11-4)$$

گام ۲- Q_0 را در معادله (۱۲-۴) قرار می دهیم و S_0 را بدست می آوریم،

$$S_k = \frac{1}{h} W_k(-A_d h Q_k) - A \quad (12-4)$$

گام ۳- بررسی می کنیم آیا همه مقادیرهای ویژه S_0 دارای قسمت های حقیقی منفی هستند یا خیر.

و به این ترتیب پایداری سیستم تعیین می شود [۳۰].

مثال ۲-۴ سیستم آونگ با پارامترهای $h = 1s$ و $\theta_0 = 20^\circ$ ، $g = 10ms^{-2}$ ، $m = 1kg$ ، $l = 2m$

را در نظر بگیرید،

$$\begin{bmatrix} \Delta\dot{\theta}(t) \\ \Delta\ddot{\theta}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} \cos(\theta_0) & 0 \end{bmatrix} \begin{bmatrix} \Delta\theta(t) \\ \Delta\dot{\theta}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{g \sin(\theta_0)}{l} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} T(t-h)$$

که در آن $\Delta\theta = \theta - \theta_0$ و

$$T(t) = [0 \quad c] \begin{bmatrix} \Delta\theta(t) \\ \Delta\dot{\theta}(t) \end{bmatrix} + mgl \sin(\theta_0)$$

می باشد. با استفاده از پارامترهای داده شده و با انتخاب مقادیرهای متفاوت برای ضریب c ، قصد داریم

به بررسی پایداری سیستم زیر پردازیم،

$$\begin{bmatrix} \Delta\dot{\theta}(t) \\ \Delta\ddot{\theta}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -4.6985 & 0 \end{bmatrix} \begin{bmatrix} \Delta\theta(t) \\ \Delta\dot{\theta}(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \frac{c}{4} \end{bmatrix} \begin{bmatrix} \Delta\theta(t-1) \\ \Delta\dot{\theta}(t-1) \end{bmatrix}$$

ابتدا رتبه A_d را بدست می آوریم. در این مثال اگر $c \neq 0$ باشد آنگاه رتبه A_d برابر یک است زیرا $eig(A_d) = \{0, \frac{c}{4}\}$ می باشد. از شاخه اصلی برای بررسی پایداری معادله دیفرانسیل تاخیری استفاده می کنیم. اگر $c = 0$ باشد آنگاه معادله دیفرانسیل تاخیری تبدیل به معادله دیفرانسیل معمولی می شود و می توان بررسی کرد سیستم پایدار مجانبی نیست زیرا $eig(A) = \pm 2.1676i$ می باشد. از طرفی چون یکی از مقدارهای ویژه A_d صفر می باشد و طبق این خاصیت از تابع لامبرت که،

$$W_k(0) = \begin{cases} 0, & k = 0 \\ -\infty, & k \neq 0 \end{cases}$$

نمی توان روش عادی که در آن شاخه های همانند ($k_1 = k_2$) استفاده می شود را برای حل مسأله به کار برد. بنابراین لازم است که روش شاخه های هیبرید^۱ ($k_1 \neq k_2$) را به کار ببریم به طوریکه در این مثال، k_2 را همیشه مساوی با صفر در نظر می گیریم.

سپس $Q_{0,0}$ را برای شاخه اصلی و با استفاده از تابع `fsolve` در نرم افزار Matlab بدست می آوریم.

فرض می کنیم،

$$Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}$$

بنابراین بدست می آوریم،

$$A_d h Q = \begin{bmatrix} 0 & 0 \\ 0 & \frac{c}{4} \end{bmatrix} \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ q_{21} \frac{c}{4} & q_{22} \frac{c}{4} \end{bmatrix}$$

همچنین می توان ماتریس های قطری شده و مُدال برای ماتریس $A_d h Q$ را به صورت زیر بدست آورد،

$$J = \begin{bmatrix} q_{22} \frac{c}{4} & 0 \\ 0 & 0 \end{bmatrix}, T = \begin{bmatrix} 0 & -\frac{q_{22}}{q_{21}} \\ 1 & 1 \end{bmatrix}$$

ماتریس تابع لامبرت نیز برای شاخه های هیبرید به صورت زیر تعریف می شود،

$$W_{k_1, k_2}(A_d h Q_{k_1, k_2}) = T \begin{bmatrix} W_{k_1}\left(q_{22} \frac{c}{4}\right) & 0 \\ 0 & W_{k_2}(0) \end{bmatrix} T^{-1}$$

^۱Hybrid Branch

بنابراین داریم،

$$W_{0,0}(A_d h Q_{0,0}) = T \begin{bmatrix} W_0 \left(q_{22} \frac{c}{4} \right) & 0 \\ 0 & W_0(0) \end{bmatrix} T^{-1}$$

که در آن ماتریس $Q_{0,0}$ در رابطه زیر صدق می کند،

$$W_{0,0}(A_d h Q_{0,0}) e^{W_{0,0}(A_d h Q_{0,0}) + Ah} = A_d h$$

بعد از محاسبه $Q_{0,0}$ می توان $S_{0,0}$ را از معادله زیر بدست آورد،

$$S_{0,0} = \frac{1}{h} W_{0,0}(A_d h Q_{0,0}) + A$$

و سپس براساس حدس مطرح شده می توان پایداری سیستم را از روی ماتریس $S_{0,0}$ تعیین کرد.

برای مثال، انتخاب می کنیم $c = 2$ و با استفاده از نرم افزار MATLAB بدست می آوریم،

$$S_{0,0} = \begin{bmatrix} 0 & 1 \\ -3.7153 & -0.1668 \end{bmatrix}$$

بنابراین مقادارهای ویژه ماتریس $S_{0,0}$ به صورت $eig(S_{0,0}) = -0.0834 \pm 1.9257i$ بدست می آید. به

این ترتیب می توان گفت با انتخاب $c = 2$ سیستم پایدار می باشد.

جدول ۲-۴ نیز نتایج را برای مقادارهای متفاوت از c نشان می دهد.

جدول ۲-۴: نتایج بدست آمده برای مقادارهای متفاوت از c با استفاده از تابع لامبرت

| c | $c = -2$ | $c = 2$ | $c = 4$ | $c = 6$ |
|-------------|---|--|--|---|
| $Q_{0,0}$ | $\begin{bmatrix} 1 & 0 \\ 1.9919 & -0.8577 \end{bmatrix}$ | $\begin{bmatrix} 1 & 0 \\ 1.6643 & -0.2823 \end{bmatrix}$ | $\begin{bmatrix} 1 & 0 \\ 1.6024 & -0.0974 \end{bmatrix}$ | $\begin{bmatrix} 1 & 0 \\ 1.5658 & 0.0148 \end{bmatrix}$ |
| $S_{0,0}$ | $\begin{bmatrix} 0 & 1 \\ -5.4264 & 0.3135 \end{bmatrix}$ | $\begin{bmatrix} 0 & 1 \\ -3.7153 & -0.1668 \end{bmatrix}$ | $\begin{bmatrix} 0 & 1 \\ -2.9122 & -0.1086 \end{bmatrix}$ | $\begin{bmatrix} 0 & 1 \\ -2.4002 & 0.0217 \end{bmatrix}$ |
| λ_0 | $0.1567 \pm 2.3242i$ | $-0.0834 \pm 1.9257i$ | $-0.0543 \pm 1.7056i$ | $0.0108 \pm 1.5492i$ |

فصل پنجم

تخصیص مقدار ویژه برای کنترل سیستم های تاخیری به وسیله تابع

لامبرت

در دهه اخیر، مسأله پایدارسازی سیستم هایی از معادلات دیفرانسیل تاخیری خطی با استفاده از کنترل پسخورد، مورد مطالعه قرار گرفته است. مسأله پایدارسازی مقاوم سیستم های تاخیری به وسیله روش تخصیص طیف متناهی در [۱۵] مورد بررسی قرار گرفته است به طوریکه در این روش سعی می شود تعداد متناهی و دلخواه از مقدارهای ویژه به موقعیت های مورد نظر انتقال داده شوند. یک روش پایدارسازی عددی نیز در [۱۶] مطرح شده است که مشابه روش تخصیص مقدار ویژه کلاسیک برای معادلات دیفرانسیل معمولی در تعیین راست ترین مقدارهای ویژه یک سیستم تاخیری می باشد. در این روش با استفاده از شبیه سازی، راست ترین مقدارهای ویژه از معادله مشخصه محاسبه می شود و سپس با استفاده از حساسیت راست ترین مقدارهای ویژه نسبت به تغییرات در بهره^۱، مقدارهای ویژه به سمت چپ صفحه مختلط انتقال پیدا می کنند.

همانطور که در فصل قبل اشاره کردیم یک روش برای پایدارسازی سیستم های تاخیری، استفاده از تابع لامبرت می باشد. در این فصل، مسأله طراحی کنترل کننده پسخورد^۲ به وسیله تخصیص مقدار ویژه برای سیستم های ناوردای زمانی خطی از معادلات دیفرانسیل تاخیری و با یک تاخیر منحصر بفرد را در نظر می گیریم. در ادامه، مثال هایی با استفاده از تابع لامبرت برای تعیین پایداری و تخصیص قطب برای یک سیستم از این معادلات ارائه می کنیم و چون برخلاف معادلات دیفرانسیل معمولی تخصیص همه مقدارهای ویژه برای یک سیستم از معادلات دیفرانسیل تاخیری به خاطر طیف نامتناهی آن عملاً شدنی نیست به همین دلیل سعی می شود راست ترین مقدارهای ویژه به محل

^۱Gain

^۲Feedback Controller

های مورد نظر انتقال داده شوند. همچنین در طراحی کنترل کننده پسخورد برای یک سیستم که به وسیله معادلات دیفرانسیل تاخیری نشان داده می شود، چون تعداد نامتناهی ماتریس S_k وجود دارد در نتیجه قرار دادن یک تعداد متناهی و انتخاب شده از مقدارهای ویژه با استفاده از روش تخصیص قطب کلاسیک برای معادلات دیفرانسیل معمولی [۳] ممکن است باعث شود سایر مقدارهای ویژه کنترل نشده به سمت راست صفحه مختلط (که اصطلاحاً نیم صفحه راست (RHP)^۱ نامیده می شود) انتقال پیدا کنند [۱۶]. اما این روش که در آن از تابع لامبرت برای کنترل یک سیستم استفاده می شود، براساس حدس مطرح شده در فصل سوم، قوانین کنترل مناسبی را بدون از دست دادن پایداری فراهم می کند.

۵-۱ تخصیص مقدار ویژه

سیستم تاخیری اسکالر خطی زیر را در نظر بگیرید،

$$\dot{x}(t) = ax(t) + a_d x(t-h) \quad (1-5)$$

جواب این سیستم با استفاده از تابع لامبرت به صورت زیر بدست می آید،

$$x(t) = \sum_{k=-\infty}^{\infty} e^{\left(\frac{1}{h} W_k(a_d h e^{-ah}) + a\right)t} C_k^I \quad (2-5)$$

معادله مشخصه این سیستم همانطور که نشان دادیم به صورت $s - a - a_d e^{-sh} = 0$ می باشد که در آن،

$$s_k = \lambda_k = \frac{1}{h} W_k(a_d h e^{-ah}) + a, k = -\infty, \dots, -1, 0, 1, \dots, \infty. \quad (3-5)$$

می باشد. همانطور که در فصل های قبل اشاره شده است، راست ترین مقدار ویژه با استفاده از شاخه اصلی بدست می آید و به این ترتیب پایداری سیستم تعیین می شود. بنابراین می توان به راست ترین مقدار ویژه مقداری در صفحه مختلط که به دلخواه انتخاب می شود را اختصاص داد.

^۱Right Half Plane

فرض می‌کنیم مقدار ویژه مورد نظر که به دلخواه انتخاب می‌شود $\lambda_{desired}$ باشد به عبارت دیگر داشته باشیم،

$$\lambda_{desired} = \lambda_0 = \frac{1}{h} W_0(a_d h e^{-ah}) + a \quad (4-5)$$

این معادله را می‌توان برای بدست آوردن پارامتر مورد نظر، با استفاده از روش‌های عددی مانند تابع fsolve در نرم افزار MATLAB حل کرد.

مثال ۱-۵ سیستم زیر را با $a = -1, h = 1s$ در نظر بگیرید،

$$\dot{x}(t) = ax(t) + a_d x(t-h)$$

مقدارهایی از a_d که مورد نیاز است برای آن که راست‌ترین مقدار ویژه به موقعیت‌های مورد نظر انتقال داده شود در جدول ۱-۵ نشان داده شده است.

جدول ۱-۵: مقدارهای متناظر از a_d برای هر مقدار ویژه مورد نظر

| | | | |
|---------------------|--------|--------|--------|
| $\lambda_{desired}$ | -0.5 | 0 | 0.5 |
| a_d | 0.3033 | 1.0000 | 2.4731 |

که این مقدارها هر کدام با استفاده از تابع fsolve در نرم افزار MATLAB و با حل معادله،

$$\lambda_{desired} = \lambda_0 = \frac{1}{h} W_0(a_d h e^{-ah}) + a$$

بدست می‌آیند. از طرفی باید توجه داشت که به خاطر محدودیت برد شاخه‌های تابع لامبرت نمی‌توان به راست‌ترین مقدارهای ویژه، هر مقداری در صفحه مختلط را اختصاص داد. برای سیستم‌های تاخیری اسکالر خطی به آسانی مشاهده می‌شود که،

$$Re\left\{\frac{1}{h} W_0(a_d h e^{-ah}) + a\right\} \geq Re\left\{\frac{-1}{h} + a\right\} \geq \frac{-1}{h} + a \quad (5-5)$$

زیرا داریم $Re\{W_0(H)\} \geq -1$ بنابراین در این مثال با استفاده از این خاصیت تابع لامبرت نتیجه می‌گیریم که،

$$Re\{\lambda_0\} = \frac{1}{h} \underbrace{Re\{W_0(a_d h e^{-ah})\}}_{\geq -1} + a \geq \frac{-1}{h} + a \geq -2$$

به همین خاطر راست ترین مقدار ویژه در این مثال نمی تواند کمتر از -2 با هر مقدار a_d باشد.

بنابراین با استفاده از مقدارهای a_d بدست آمده، راست ترین مقدارهای ویژه به موقعیت های مورد نظر انتقال داده می شوند. با محاسبه،

$$\lambda_k = \frac{1}{h} W_k(a_d h e^{-ah}) + a$$

برای $k = -1, 0, 1$ و $a_d = 0.3033$ با استفاده از نرم افزار MATLAB بدست می آوریم،

جدول ۲-۵: مقادیر ویژه محاسبه شده برای سیستم در مثال ۱-۵ با $a_d = 0.3033$

| |
|------------------------------------|
| $\lambda_{-1} = -2.7333 - 4.3318i$ |
| $\lambda_0 = -0.5$ |
| $\lambda_1 = -2.7333 + 4.3318i$ |

همچنین برای $k = -3, \dots, 3$ و $a_d = 1.0000$ خواهیم داشت،

جدول ۳-۵: مقادیر ویژه محاسبه شده برای سیستم در مثال ۱-۵ با $a_d = 1.0000$

| | |
|----------------------------------|-------------------------------------|
| $\lambda_0 = 0$ | |
| $\lambda_1 = -1.5321 + 4.5972i$ | $\lambda_{-1} = -1.5321 - 4.5972i$ |
| $\lambda_2 = -2.3940 + 10.688i$ | $\lambda_{-2} = -2.3940 - 10.688i$ |
| $\lambda_3 = -2.8490 + 17.1715i$ | $\lambda_{-3} = -2.8490 - 17.1715i$ |

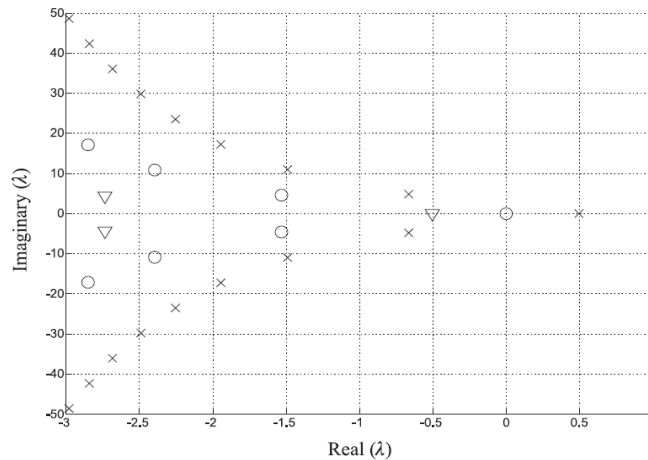
و برای $k = -7, \dots, 7$ و $a_d = 2.4731$ نیز بدست می آوریم،

جدول ۴-۵: مقادیر ویژه محاسبه شده برای سیستم در مثال ۱-۵ با $a_d = 2.4731$

| | |
|----------------------------------|-------------------------------------|
| $\lambda_0 = 0.5$ | |
| $\lambda_1 = -0.6621 + 4.7829i$ | $\lambda_{-1} = -0.6621 - 4.7829i$ |
| $\lambda_2 = -1.4890 + 10.9510i$ | $\lambda_{-2} = -1.4890 - 10.9510i$ |
| $\lambda_3 = -1.9423 + 17.2241i$ | $\lambda_{-3} = -1.9423 - 17.2241i$ |
| $\lambda_4 = -2.2533 + 23.5087i$ | $\lambda_{-4} = -2.2533 - 23.5087i$ |
| $\lambda_5 = -2.4901 + 29.7952i$ | $\lambda_{-5} = -2.4901 - 29.7952i$ |
| $\lambda_6 = -2.6814 + 36.0817i$ | $\lambda_{-6} = -2.6814 - 36.0817i$ |
| $\lambda_7 = -2.8419 + 42.3681i$ | $\lambda_{-7} = -2.8419 - 42.3681i$ |

شکل ۱-۵ نیز طیف ویژه این سیستم را نشان می دهد. همانطور که مشاهده می شود با استفاده از مقدارهای a_d بدست آمده می توان راست ترین مقدارهای ویژه را دقیقاً به موقعیت های مورد نظر

انتقال داد. $(-0.5(\nabla); 0(O); 0.5(\times))$



شکل ۱-۵: طیف ویژه سیستم (۱-۵) با $a = -1, h = 1s$ و a_d های بدست آمده در جدول ۱-۵ [۲۷]

برای سیستم های تاخیری از مرتبه بالاتر مسأله پیچیده می شود و با توجه به ساختار و پارامترهای یک سیستم محدودیت هایی در تخصیص مقدار ویژه وجود دارد به طوریکه اگر مقدارهای نادرست اختصاص داده شود روش در رسیدن به جواب با شکست مواجه خواهد شد. همچنین تعیین دقیق یک محدوده می تواند مشکل باشد. اما می توان با روش تکراری آزمون و خطا این کار را انجام داد. برای مثال اگر مقدار انتخاب شده مناسب نباشد مقدارهای کوچکتر از مقدار مورد نظر و یا مقدارهای متفاوت از راست ترین مقدارهای ویژه مورد نظر را انتخاب می کنیم و مسأله را دوباره حل می کنیم.

۲-۵ طراحی کنترل کننده پسخورد

۱-۲-۵ حالت اسکالر

معادله دیفرانسیل تاخیری اسکالر زیر را در نظر بگیرید،

$$\dot{x}(t) = ax(t) + a_d x(t-h) + u(t), u(t) = kx(t) \quad (۶-۵)$$

معادله مشخصه این سیستم به صورت،

$$s - a - a_d e^{-sh} - k = 0 \quad (۷-۵)$$

می باشد. با استفاده از روش تقریبات پاده به جای عبارت نمایی در معادله مشخصه، کسر زیر را جایگزین می کنیم،

$$e^{-sh} = \frac{e^{-\frac{sh}{2}}}{e^{\frac{sh}{2}}} \approx \frac{1 - \frac{sh}{2}}{1 + \frac{sh}{2}} \quad (۸-۵)$$

رابطه (۸-۵) را در معادله (۷-۵) قرار می دهیم و بدست می آوریم،

$$s^2 h + s(2 - ah - kh + a_d h) - 2(a + k) - 2a_d = 0 \quad (۹-۵)$$

به این ترتیب برای یک قطب مورد نظر می توان بهره کنترل k را بدست آورد. برای مثال با در نظر گرفتن پارامترهای $a = 1, a_d = -2, h = 1s$ به صورت زیر بدست می آید،

$$s^2 - s(1 + k) + 2 - 2k = 0 \quad (۱۰-۵)$$

از تئوری چند جمله ای ها می دانیم شرط لازم برای آن که معادله (۱۰-۵) دارای دو قطب پایدار باشد آن است که ضرایب آن هم علامت و هیچکدام صفر نباشند [۳۴]. بنابراین در این مثال باید داشته باشیم $k < -1$ که با قرار دادن k مساوی -1.1 نتیجه می گیریم معادله دارای دو قطب پایدار به صورت زیر است،

$$s_1, s_2 = -0.500 \pm 2.0488i$$

با قرار دادن این بهره کنترل به همراه پارامترهای $a = 1, a_d = -2, h = 1s$ در معادله سیستم (۶-۵) خواهیم داشت،

$$\dot{x}(t) = -0.1x(t) - 2x(t - 1)$$

معادله مشخصه این سیستم به صورت $s + 0.1 + 2e^{-s} = 0$ می باشد به دلیل وجود عبارت نمایی در معادله مشخصه از تابع لامبرت استفاده می کنیم و جواب های این معادله مشخصه را به صورت زیر محاسبه می کنیم،

$$s_k = W_k(-2e^{0.1}) - 0.1$$

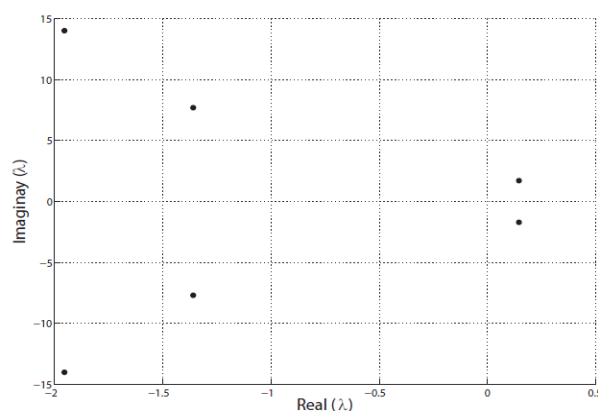
برای $k = -2, -1, 0, \dots, 3$ و با استفاده از نرم افزار MATLAB به محاسبه مقادیر ویژه می پردازیم و

بدست می آوریم،

جدول ۵-۵: مقادیر ویژه محاسبه شده برای سیستم (۳۴-۵) با $\mathbf{h} = \mathbf{1}s$ ، $\mathbf{a}_d = -2$ ، $\mathbf{a} = \mathbf{1}$ و $k = -1.1$

| | |
|----------------------------|-------------------------------|
| $s_0 = 0.1449 + 1.7128i$ | $s_{-1} = 0.1449 - 1.7128i$ |
| $s_1 = -1.3602 + 7.6916i$ | $s_{-2} = -1.3602 - 7.6916i$ |
| $s_2 = -1.9550 + 14.0055i$ | $s_{-3} = -1.9550 - 14.0055i$ |

طیف ویژه این سیستم در شکل ۲-۵ نیز نشان داده شده است.



شکل ۲-۵: طیف ویژه با کنترل کننده پسخورد طراحی شده با استفاده از تقریب پاده [۲۷]

طیف ویژه با کنترل کننده پسخورد طراحی شده با استفاده از روش تقریبات پاده، شکست انتقال قطب به موقعیت مورد نظر و بنابراین پایدار کردن سیستم را نشان می دهد.

اگر چه تقریب های پاده مرتبه بالاتر یا تقریب های کسری پیشرفته می تواند برای تقریب عبارت نمایی در معادله مشخصه استفاده شود اما چنین روش هایی محدودیت هایی در دقت مسأله ایجاد می کنند و ممکن است منجر به ناپایداری سیستم اصلی شوند [۲۰، ۲۳].

همچنین از معادله مشخصه سیستم (۶-۵) با قطب مورد نظر $\lambda_{desired}$ ، معادله خطی برای k به صورت زیر بدست می آید،

$$\lambda_{desired} - a - a_d e^{-\lambda_{desired} h} - k = 0 \quad (11-5)$$

برای مثال با در نظر گرفتن پارامترهای $a = 1, a_d = -1, h = 1s$ و با جایگزین کردن $\lambda_{desired} = -1$ در معادله (۵-۱۱) بهره $k = 0.7183$ را بدست می آوریم. سپس با به کار بردن این بهره کنترل برای سیستم (۵-۶) بدست می آوریم،

$$\dot{x}(t) = 1.7183x(t) - x(t - 1)$$

معادله مشخصه این سیستم به صورت،

$$s - 1.7183 + e^{-s} = 0$$

می باشد. ریشه های این معادله را از رابطه،

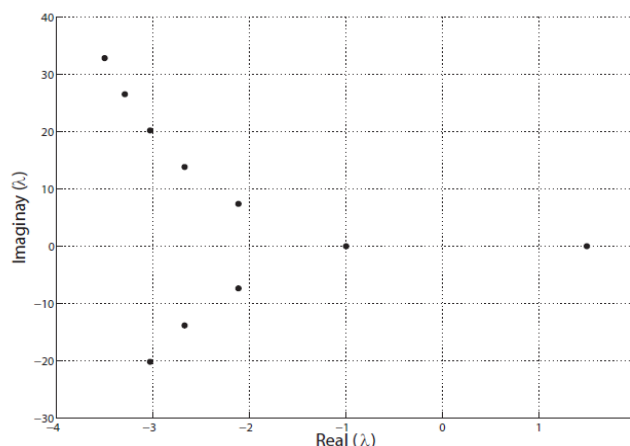
$$s_k = W_k(-e^{-1.7183}) + 1.7183$$

محاسبه می کنیم. به این ترتیب برای $k = -4, \dots, 0, \dots, 4$ و با استفاده از نرم افزار MATLAB مقادیر ویژه را به صورت زیر بدست می آوریم،

جدول ۵-۶: مقادیر ویژه محاسبه شده برای سیستم (۵-۳۴) با $a = 1, a_d = -1, h = 1s$ و $k = 0.7183$

| $s_0 = 1.4938$ | |
|----------------------------|-------------------------------|
| $s_1 = -1.000$ | $s_{-1} = -1.000$ |
| $s_2 = -2.1177 + 7.3743i$ | $s_{-2} = -2.1177 - 7.3743i$ |
| $s_3 = -2.6749 + 13.8296i$ | $s_{-3} = -2.6749 - 13.8296i$ |
| $s_4 = -3.021 + 20.1893i$ | $s_{-4} = -3.021 - 20.1893i$ |

طیف ویژه این سیستم در شکل ۵-۳ نشان داده شده است.



شکل ۵-۳: طیف ویژه با کنترل کننده پسخورد طراحی شده با استفاده از روش خطی [۲۷]

مشاهده می شود زمانی که یکی از قطب ها در موقعیت مطلوب 1- قرار گیرد راست ترین قطب در سمت راست صفحه مختلط می باشد و اگر چه برای مقدار ویژه مورد نظر، بهره k بدست می آید اما هنگامی که بهره بدست آمده را در معادله (۵-۱۱) به کار می بریم تعداد نامتناهی قطب وجود دارد که در این معادله صدق می کنند و برخی از آن ها دارای قسمت های حقیقی بزرگتر از قطب مورد نظر هستند. به این ترتیب مشابه روش قبل، طیف ویژه با کنترل کننده پسخورد طراحی شده با استفاده از روش خطی نیز شکست انتقال قطب به موقعیت مورد نظر و در نتیجه پایدار کردن سیستم را نشان می دهد. در ادامه نشان داده می شود که با استفاده از تابع لامبرت می توان قسمت حقیقی راست ترین قطب را دقیقاً به موقعیت مورد نظر انتقال داد و سیستم را پایدار کرد.

سیستم زیر را در نظر بگیرید،

$$\dot{x}(t) = ax(t) + a_d x(t-h) + u(t), u(t) = kx(t) \quad (۱۲-۵)$$

معادله مشخصه این سیستم به صورت زیر می باشد،

$$s - a - a_d e^{-sh} - k = 0 \quad (۱۳-۵)$$

ریشه های معادله مشخصه یا همان مقادیر ویژه را با استفاده از تابع لامبرت به صورت زیر محاسبه می کنیم،

$$s_k = \frac{1}{h} W_k(a_d h e^{-(a+k)h}) + a + k \quad (۱۴-۵)$$

حال در معادله سیستم (۵-۶) قرار می دهیم $a = 1, a_d = -1, h = 1s$ و مقدار ویژه مورد نظر را مساوی با 1- در نظر می گیریم و با استفاده از تابع لامبرت k را به گونه ای پیدا می کنیم که،

$$Re\left\{s_0 = \frac{1}{h} W_0(a_d h e^{-(a+k)h}) + a + k\right\} = -1$$

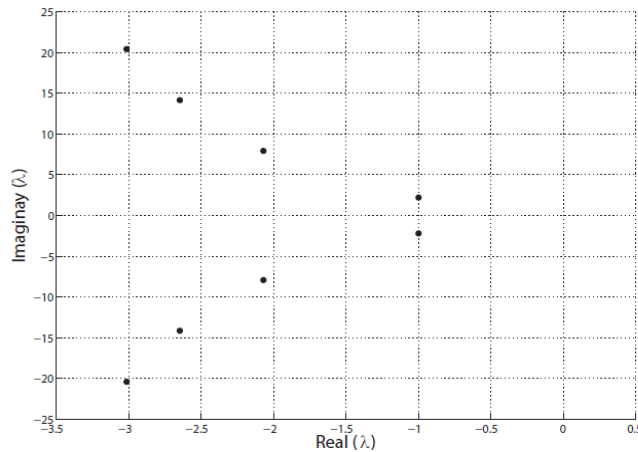
باشد که با حل این معادله با استفاده از تابع fsolve در نرم افزار MATLAB، $k = -3.5978$ بدست می آید.

همچنین برای $k = -4, \dots, 0, \dots, 4$ سایر مقادیرهای ویژه به صورت زیر خواهند بود،

جدول ۷-۵: مقادیر ویژه محاسبه شده برای سیستم (۳۴-۵) با $h = 1s$ ، $a_d = -1$ ، $a = 1$ و $k = -3.5978$

| | |
|----------------------------|-------------------------------|
| $s_0 = -1 - 2.1991i$ | |
| $s_1 = -1 + 2.1991i$ | $s_{-1} = -1 - 2.1991i$ |
| $s_2 = -2.0716 + 7.9203i$ | $s_{-2} = -2.0716 - 7.9203i$ |
| $s_3 = -2.6486 + 14.1336i$ | $s_{-3} = -2.6486 - 14.1336i$ |
| $s_4 = -3.0157 + 20.3999i$ | $s_{-4} = -3.0157 - 20.3999i$ |

شکل ۴-۵ نیز طیف ویژه سیستم را با بهره کنترل بدست آمده، نشان می دهد.



شکل ۴-۵: طیف ویژه با کنترل کننده پسخورد طراحی شده با استفاده از روش تابع لامبرت [۲۷]

طیف ویژه با کنترل کننده پسخورد طراحی شده با استفاده از تابع لامبرت، نشان می دهد راست ترین مقدار ویژه دقیقاً در موقعیت مورد نظر قرار گرفته است.

مثال ۲-۵ (سیستم هیدرولیکی) یک مدل خطی برای سیستم مخزن آب هیدرولیکی در نظر بگیرید،

$$\dot{x}(t) = \frac{1}{A\rho} \left(u_{in}(t-h) - \frac{\rho g x(t)}{R} \right)$$

که در آن $A = 10m^2$ سطح مقطع عرضی است، $\rho = 1000kgm^{-3}$ چگالی آب، $g = 10ms^{-2}$ شتاب گرانشی، $h = 1s$ تاخیر، t زمان، $R = 10m^{-\frac{1}{2}}kg^{-\frac{1}{2}}$ مقاومت هیدرو دینامیکی لوله خروجی، u_{in} سرعت جریان ورودی و $x(t)$ سطح تراز آب می باشد. در این مثال قصد داریم یک کنترل پسخورد مناسب

$u_{in} = kx(t)$ طراحی کنیم و قطب های سیستم حلقه بسته را به موقعیت های خاص انتقال دهیم. ابتدا یادآوری می کنیم که سیستم حلقه بسته می تواند به وسیله معادلات دیفرانسیل تاخیری و به صورت زیر نشان داده شود،

$$\dot{x}(t) = -\frac{g}{AR}x(t) + \frac{k}{A\rho}x(t-h) = -0.1x(t) - 0.1x(t-1)$$

بنابراین بدست می آوریم،

$$\lambda_k = \frac{1}{h}W_k \left(\frac{k}{A\rho} h e^{\frac{g}{AR}h} \right) - \frac{g}{AR}, k = -\infty, \dots, -1, 0, 1, \dots, \infty$$

همانطور که می دانیم برای حالت اسکالر ثابت شده است که راست ترین قطب با استفاده از شاخه اصلی تابع لامبرت بدست می آید. همچنین توجه داریم که هر شاخه از تابع لامبرت دارای برد مخصوص به خود است به طوریکه برای شاخه اصلی $Re\{W_0(H)\} \geq -1$ می باشد. پس در این مثال داریم،

$$Re \left\{ \frac{1}{h}W_0 \left(\frac{k}{A\rho} h e^{\frac{g}{AR}h} \right) - \frac{g}{AR} \right\} \geq Re \left\{ -\frac{1}{h} - \frac{g}{AR} \right\} \geq -\frac{1}{h} - \frac{g}{AR} \geq -1.1$$

بنابراین نمی توان راست ترین مقدار ویژه مورد نظر را کمتر از -1.1 انتخاب کرد زیرا شرایط برای تخصیص مقدار ویژه به طور دلخواه برای سیستم هایی از معادلات دیفرانسیل تاخیری وجود ندارد و برای حالت اسکالر نیز با توجه به مقدار تاخیر زمانی و ضرایب سیستم، محدودیت در تخصیص مقدار ویژه وجود دارد (ضمیمه B را ببینید). برای مثال، راست ترین مقدار ویژه را در موقعیت -0.5 قرار دهیم و k را متناظر با مقدار ویژه ای که در نظر می گیریم از حل معادله،

$$\lambda_{desired} = Re \left\{ \frac{1}{h}W_0 \left(\frac{k}{A\rho} h e^{\frac{g}{AR}h} \right) - \frac{g}{AR} \right\}$$

و با استفاده از روش های عددی مانند تابع fsolve در نرم افزار MATLAB بدست می آوریم به طوریکه برای این مثال $k = -2.4261$ بدست می آید.

۵-۲-۲ سیستم‌هایی با تاخیر در ورودی

سیستم زیر از معادلات دیفرانسیل معمولی با تاخیر در ورودی را در نظر بگیرید،

$$\dot{x}(t) = Ax(t) + Bu(t-h) \quad (15-5)$$

$$u(t) = Kx(t)$$

بنابراین سیستم حلقه بسته به یک سیستم از معادلات دیفرانسیل تاخیری به صورت زیر تبدیل می‌شود،

$$\dot{x}(t) = Ax(t) + BKx(t-h) \quad (16-5)$$

الگوریتم زیر روش محاسباتی بدست آوردن بهره K را در حالت کلی شرح می‌دهد.

الگوریتم ۵-۱

گام ۱- ابتدا مقادارهای ویژه مورد نظر $\lambda_{i,desired}$ را برای $i = 1, \dots, n$ انتخاب می‌کنیم و قرار می‌دهیم،

$$\lambda_i(S_0) = \lambda_{i,desired} \quad (17-5)$$

که در آن $\lambda_i(S_0)$ ، i امین مقدار ویژه از ماتریس S_0 است.

گام ۲- دو ماتریس ضرایب جدید $A = A_d, A_d = BK$ را در معادله (۱۸-۵) قرار می‌دهیم،

$$W_k(A_d h Q_k) e^{W_k(A_d h Q_k) + A_d h} = A_d h \quad (18-5)$$

و به روش عددی ماتریس Q_0 را بدست می‌آوریم. توجه کنید که K یک ماتریس مجهول و با همه درایه‌های نامعلوم می‌باشد و Q_0 یک تابع از مجهول K می‌باشد.

گام ۳- ماتریس Q_0 بدست آمده را در معادله (۱۹-۵) قرار می‌دهیم،

$$S_k = \frac{1}{h} W_k(A_d h Q_k) + A \quad (19-5)$$

سپس S_0 و مقادارهای ویژه آن را به صورت تابعی از ماتریس مجهول K محاسبه می‌کنیم.

گام ۴- معادله (۱۷-۵) را با استفاده از روش های عددی مانند fsolve در نرم افزار MATLAB حل می کنیم [۲۷].

همچنین روش تخصیص قطب برای سیستم حلقه بسته (۱۶-۵) را می توان به طور خلاصه و با استفاده از الگوریتم زیر بیان کرد.

الگوریتم ۲-۵

گام ۱- ابتدا راست ترین مقدارهای ویژه $\lambda_{i,desired}$ را برای $i = 1, \dots, n$ انتخاب می کنیم.

گام ۲- شرط اولیه $K = K_0$ را در نظر می گیریم.

گام ۳- تا زمانی که $\lambda_i(S_{0,new}) - \lambda_{i,desired} \geq tolerance$ و برای $i = 1, \dots, n$

۱- انتخاب می کنیم $K = K_{new}$

۲- فرض می کنیم $A_{d,new} = BK_{new}$ معادله زیر را برای $Q_{0,new}$ حل می کنیم به طوریکه،

$$W_0(A_{d,new}hQ_{0,new})e^{(W_0(A_{d,new}hQ_{0,new})+Ah)} = A_{d,new}h$$

۳- محاسبه می کنیم،

$$S_{0,new} = \frac{1}{h}W_0(A_{d,new}hQ_{0,new}) + A$$

مثال ۳-۵ معادله ون در پل^۱ زیر را در نظر بگیرید،

$$\ddot{x}(t) + f(x, t)\dot{x}(t) + x(t) = g(x, t; h)$$

که در آن،

$$f(x, t) = \varepsilon(x^2(t) - 1)$$

طرف چپ این معادله می تواند به صورت زیر نیز نشان داده شود،

^۱Van Der Pol

$$g(x, t; h) = k_1 x(t-h) + k_2 \dot{x}(t-h)$$

با قرار دادن هر کدام از این عبارت ها در معادله ون در پل بدست می آوریم،

$$\ddot{x}(t) + x(t) = \varepsilon(1 - x^2(t))\dot{x}(t) + k_1 x(t-h) + k_2 \dot{x}(t-h)$$

با خطی کردن معادله بالا در نقطه تعادل صفر و برای اغتشاش های بینهایت کوچک داریم،

$$\ddot{x}(t) + x(t) = \varepsilon \dot{x}(t) + k_1 x(t-h) + k_2 \dot{x}(t-h)$$

یا به طور معادل و با تعریف $x_1 = x$ و $x_2 = \dot{x}$ معادله های حالت زیر را بدست می آوریم،

$$\begin{cases} \dot{x}_1(t) \\ \dot{x}_2(t) \end{cases} = \begin{bmatrix} 0 & 1 \\ -1 & \varepsilon \end{bmatrix} \begin{cases} x_1(t) \\ x_2(t) \end{cases} + \begin{bmatrix} 0 & 0 \\ k_1 & k_2 \end{bmatrix} \begin{cases} x_1(t-h) \\ x_2(t-h) \end{cases} \quad (20-5)$$

یا

$$\dot{x}(t) = \underbrace{\begin{bmatrix} 0 & 1 \\ -1 & \varepsilon \end{bmatrix}}_A x(t) + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_B \underbrace{\begin{bmatrix} k_1 & k_2 \end{bmatrix}}_K x(t-h)$$

قبل از تخصیص مقدار ویژه برای بدست آوردن کنترل کننده پسخورد در این مثال به بررسی کنترل پذیری این سیستم می پردازیم.

در طی دهه اخیر شرایط برای کنترل پذیری سیستم،

$$\dot{x}(t) = Ax(t) + Bu(t-h) \quad (21-5)$$

توسط آلبرات^۱ در [۱۷] مطرح شد به گونه ای که سیستم روی بازه $[0, t_1]$ کنترل پذیر است اگر یک کنترل $u(t)$ قابل قبول (یعنی اندازه پذیر و کراندار روی یک بازه زمانی متناهی) وجود داشته باشد به قسمی که $x(t_1) = 0, t_1 > h$.

همچنین مطابق با معیار ارائه شده توسط آلبرات، سیستم روی بازه $[0, t_1]$ کنترل پذیر است اگر و تنها اگر،

$$\text{rank}[B : AB] = n \quad (22-5)$$

^۱Olbrot

باشد. بنابراین سیستم در این مثال، کنترل پذیر است زیرا،

$$\text{rank}[B : AB] = 2$$

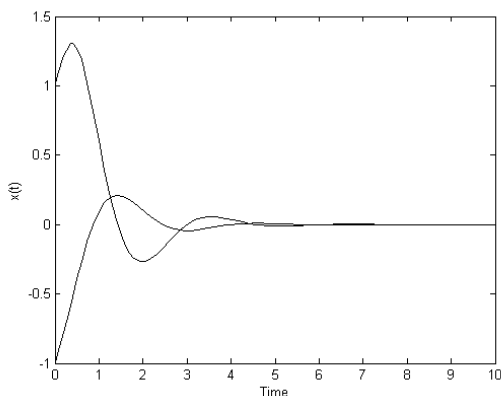
می باشد. به این ترتیب با استفاده از روش جابجایی قطب^۱ (تخصیص مقدار ویژه) که در قسمت قبل معرفی شد، می توان کنترل کننده پسخورد مناسب برای پایدار کردن سیستم طراحی کرد و با انتخاب بهره های k_1 و k_2 ، مقدارهای ویژه را در موقعیت های مورد نظر در صفحه مختلط قرار داد.

حال اگر در معادله (۵-۲۰) داشته باشیم $k_1 = k_2 = 0$ آنگاه سیستم زمانی که $\varepsilon = 0.1$ باشد ناپایدار است و مقدارهای ویژه آن $0.0500 \pm 0.9987i$ هستند.

برای مثال زمانی که $h = 0.2s$ باشد اگر مقدارهای ویژه مورد نظر $-1 \pm 2i$ باشند که به طور دلخواه انتخاب می شوند آنگاه با استفاده از نرم افزار MATLAB و تابع `place_dde` بدست می آوریم،

$$k_1 = -1.9667, k_2 = -1.8773$$

شکل ۵-۵ پاسخ این سیستم را بعد از به کار بردن پسخورد $K = [-1.9667 \quad -1.8773]$ نشان می دهد. همانطور که مشاهده می شود پسخورد بدست آمده سیستم را پایدار می کند.



شکل ۵-۵: پاسخ سیستم بعد از به کار بردن پسخورد $K = [-1.9667 \quad -1.8773]$

مثال ۴-۵ (سیستم آونگ) سیستم آونگ با پارامترهای $l = 2m$ ، $m = 1kg$ ، $g = 10ms^{-2}$ ،

^۱Pole Placement

که در آن $\Delta\theta = \theta - \theta_0$ و $h = 0.2s$ را در نظر بگیرید،

$$\begin{bmatrix} \Delta\dot{\theta}(t) \\ \Delta\ddot{\theta}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l}\cos(\theta_0) & 0 \end{bmatrix} \begin{bmatrix} \Delta\theta(t) \\ \Delta\dot{\theta}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{g\sin(\theta_0)}{l} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} T(t-h)$$

بنابراین با استفاده از پارامترهای داده شده، سیستم حلقه بسته به صورت،

$$T(t) = [k_1 \quad k_2] \begin{bmatrix} \Delta\theta(t) \\ \Delta\dot{\theta}(t) \end{bmatrix} + mgl\sin(\theta_0)$$

و یا به صورت،

$$\begin{bmatrix} \Delta\dot{\theta}(t) \\ \Delta\ddot{\theta}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -4.6985 & 0 \end{bmatrix} \begin{bmatrix} \Delta\theta(t) \\ \Delta\dot{\theta}(t) \end{bmatrix} + \begin{bmatrix} \frac{0}{4} & \frac{0}{4} \\ \frac{k_1}{4} & \frac{k_2}{4} \end{bmatrix} \begin{bmatrix} \Delta\theta(t-0.2) \\ \Delta\dot{\theta}(t-0.2) \end{bmatrix}$$

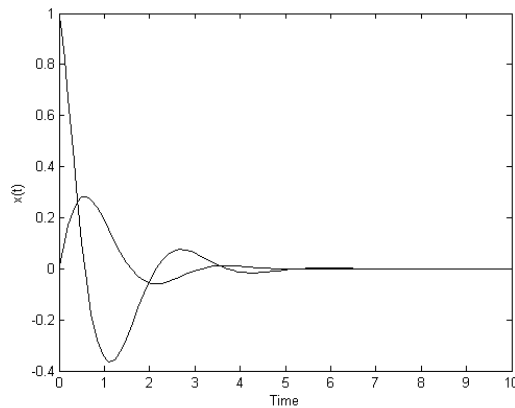
نشان داده می شود. این سیستم کنترل پذیر است زیرا،

$$\begin{bmatrix} \Delta\dot{\theta}(t) \\ \Delta\ddot{\theta}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -4.6985 & 0 \end{bmatrix} \begin{bmatrix} \Delta\theta(t) \\ \Delta\dot{\theta}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{4} \end{bmatrix} [k_1 \quad k_2] \begin{bmatrix} \Delta\theta(t-0.2) \\ \Delta\dot{\theta}(t-0.2) \end{bmatrix}$$

بنابراین هدف، پیدا کردن $K = [k_1 \quad k_2]$ به گونه ای است که راست ترین قطب های سیستم حلقه بسته در موقعیت $-1 \pm 2i$ قرار بگیرند.

$$\text{rank}[B : AB] = 2$$

با استفاده از نرم افزار MATLAB، $k_1 = 5.1845$ ، $k_2 = -4.9190$ را بدست می آوریم. شکل ۵-۶ نیز پاسخ سیستم را بعد از به کار بردن این پسخورد نشان می دهد.



شکل ۵-۶: پاسخ سیستم بعد از به کار بردن پسخورد $K = [5.1845 \quad -4.9190]$

۳-۲-۵ سیستم هایی با تاخیر در متغیر حالت

سیستم زیر از معادلات دیفرانسیل با تاخیر در متغیر حالت را در نظر بگیرید،

$$\dot{x}(t) = Ax(t) + A_d x(t-h) + Bu(t) \quad (23-5)$$

با در نظر گرفتن $u(t) = Kx(t) + K_d x(t-h)$ سیستم حلقه بسته به صورت زیر بدست می آید،

$$\dot{x}(t) = (A + BK)x(t) + (A_d + BK_d)x(t-h) \quad (24-5)$$

بهره های K, K_d برای این سیستم با تخصیص مقدار ویژه و با استفاده از الگوریتم زیر تعیین می شوند.

الگوریتم ۳-۵

گام ۱- ابتدا مقادیر ویژه مورد نظر $\lambda_{i,desired}$ را برای $i = 1, \dots, n$ انتخاب می کنیم و قرار می دهیم،

$$\lambda_i(S_0) = \lambda_{i,desired} \quad (25-5)$$

که در آن $\lambda_i(S_0)$ ، i امین مقدار ویژه از S_0 است.

گام ۲ - دو ماتریس ضرایب جدید $A = A + BK, A_d = A_d + BK_d$ را در معادله (۲۶-۵) به کار می

بریم،

$$W_k(A_d h Q_k) e^{W_k(A_d h Q_k) + A h} = A_d h \quad (26-5)$$

و این معادله را برای بدست آوردن Q_0 و برای شاخه اصلی حل می کنیم.

توجه کنید که K, K_d ماتریس های مجهول و با همه درایه های نامعلوم می باشند و Q_0 یک تابع از مجهول K, K_d می باشد.

گام ۳- ماتریس Q_0 بدست آمده را در معادله (27-5) قرار می دهیم،

$$S_k = \frac{1}{h} W_k(A_d h Q_k) + A \quad (27-5)$$

سپس S_0 و مقدارهای ویژه آن را به صورت تابعی از ماتریس مجهول K, K_d محاسبه می کنیم.

گام ۴- سپس معادله (28-5) را با استفاده از روش های عددی مانند fsolve در نرم افزار MATLAB حل می کنیم [27].

همچنین روش تخصیص قطب برای سیستم حلقه بسته (29-5) را می توان به طور خلاصه و با استفاده از الگوریتم زیر نیز بیان کرد.

الگوریتم ۴-۵

گام ۱- ابتدا راست ترین مقدارهای ویژه $\lambda_{i,desired}$ را برای $i = 1, \dots, n$ انتخاب می کنیم.

گام ۲- شرایط اولیه $K = K_0$ و $K_d = K_{d0}$ را در نظر می گیریم.

گام ۳- تا زمانی که $\lambda_i(S_{0,new}) - \lambda_{i,desired} \geq tolerance$ و برای $i = 1, \dots, n$

۱- انتخاب می کنیم $K = K_{new}$ و $K_d = K_{d,new}$

۲- فرض می کنیم $A_{new} = A + BK$ و $A_{d,new} = A_d + BK_d$ معادله زیر را برای

$Q_{0,new}$ حل می کنیم به قسمی که،

$$W_0(A_{d,new} h Q_{0,new}) e^{W_0(A_{d,new} h Q_{0,new}) + A_{new} h} = A_{d,new} h$$

۳- محاسبه می کنیم،

$$S_{0,new} = \frac{1}{h} W_0(A_{d,new} h Q_{0,new}) + A_{new}$$

بررسی کنترل پذیری چنین سیستمی با استفاده از فرم جواب بدست آمده از تابع لامبرت در

[۲۹] مورد مطالعه قرار گرفت که در ادامه خلاصه ای از آن را بیان می کنیم.

سیستم زیر را در نظر بگیرید،

$$\dot{x}(t) = Ax(t) + A_d x(t-h) + Bu(t), t > 0$$

$$x(t) = g(t), t \in [-h, 0) \quad (28-5)$$

$$x(t) = x_0, t = 0$$

تعریف ۱-۵ سیستم (۲۸-۵) کنترل پذیر نقطه ای^۱ است اگر برای هر شرط اولیه $x_0, g(t)$ یک زمان

$t_1, 0 < t_1 < \infty$ و یک کنترل قابل قبول (اندازه پذیر و کراندار در یک بازه زمانی متناهی) $u(t)$ برای

$t \in [0, t_1]$ وجود داشته باشد به قسمی که [۲۹]،

$$x(t_1; g, x_0, u(t)) = 0 \quad (29-5)$$

در ادامه سه شرط لازم و کافی برای کنترل پذیری نقطه ای که معادل هم می باشند را بیان می کنیم.

سیستم (۲۸-۵) کنترل پذیر نقطه ای است اگر و تنها اگر،

$$\text{rank} [C(0, t_1) = \int_0^{t_1} \sum_{k=-\infty}^{\infty} e^{S_k(t_1-\xi)} C_k^N B B^T \left\{ \sum_{k=-\infty}^{\infty} e^{S_k(t_1-\xi)} C_k^N \right\}^T d\xi] = n \quad -1$$

۲- همه ی سطرهای $\sum_{k=-\infty}^{\infty} e^{S_k(t-0)} C_k^N B$ در بازه $[0, \infty)$ مستقل خطی باشند.

۳- همه ی سطرهای ماتریس $(sI - A - A_d e^{-sh})^{-1} B$ بر روی میدان اعداد مختلط به جز در ریشه

های معادله مشخصه، مستقل خطی باشند.

^۱Point-Wise Controllable

شرط سوم برای کنترل پذیری نقطه ای را می توان با محاسبه تبدیل لاپلاس $\sum_{k=-\infty}^{\infty} e^{S_k(t-0)} C_k^N B$ بدست آورد. برای پیدا کردن این شرط از معادله،

$$\dot{x}(t) = Ax(t) + A_d x(t-h) + Bu(t) \quad (30-5)$$

به صورت زیر تبدیل لاپلاس می گیریم،

$$sX(s) - x(0) = AX(s) + A_d e^{-sh} X(s) + A_d \int_0^h e^{-st} g(t-h) dt + BU(s)$$

بنابراین،

$$X(s) = (sI - A - A_d e^{-sh})^{-1} \left(x(0) + A_d \int_0^h e^{-st} g(t-h) dt \right) + (sI - A - A_d e^{-sh})^{-1} BU(s)$$

به این ترتیب خواهیم داشت،

$$x(t) = \mathcal{L}^{-1} \left((sI - A - A_d e^{-sh})^{-1} \left(x(0) + A_d \int_0^h e^{-st} g(t-h) dt \right) \right) + \mathcal{L}^{-1} \left((sI - A - A_d e^{-sh})^{-1} BU(s) \right) \quad (31-5)$$

با مقایسه این رابطه با جواب سیستم (28-5) که با استفاده از تابع لامبرت و به صورت زیر بدست می آید،

$$x(t) = \sum_{k=-\infty}^{\infty} e^{S_k t} C_k^I + \int_0^t \sum_{k=-\infty}^{\infty} e^{S_k(t-\xi)} C_k^N B u(\xi) d\xi \quad (32-5)$$

نتیجه می گیریم که،

$$\mathcal{L} \left(\sum_{k=-\infty}^{\infty} e^{S_k(t-0)} C_k^N B \right) = (sI - A - A_d e^{-sh})^{-1} B \quad (33-5)$$

و چون تبدیل لاپلاس، یک عملگر خطی یک به یک است در نتیجه شرط سوم برای کنترل پذیری بدست می آید.

برای مشاهده اثبات کامل معادل بودن این سه شرط به [29] رجوع شود.

به این ترتیب اگر یک سیستم، کنترل پذیر نقطه ای باشد طراحی کنترل کننده پسخورد خطی به وسیله تخصیص مقدار ویژه برای آن امکان پذیر است و در غیر اینصورت امکان پذیر نمی باشد.

مثال ۵-۵ سیستم زیر را در نظر بگیرید،

$$\dot{x}(t) = \begin{bmatrix} 1.1 & -0.1732 \\ -0.0577 & 1.1 \end{bmatrix} x(t) + \begin{bmatrix} 0.35 & 0.2598 \\ 0.0866 & 0.35 \end{bmatrix} x(t-h) + \begin{bmatrix} 1 \\ -0.5774 \end{bmatrix} u(t)$$

برای بررسی کنترل پذیری سیستم از این شرط استفاده می کنیم که سیستم، کنترل پذیر نقطه ای است اگر و تنها اگر سطرهای ماتریس $(sI - A - A_d e^{-sh})^{-1} B$ مستقل خطی باشند. به این ترتیب بدست می آوریم،

$$(sI - A - A_d e^{-sh})^{-1} B = \begin{bmatrix} 5 \\ \frac{5s - 6 - e^{-sh}}{-5/\sqrt{3}} \\ 5s - 6 - e^{-sh} \end{bmatrix}$$

واضح است که دو سطر ماتریس، وابسته خطی هستند بنابراین سیستم، کنترل پذیر نقطه ای نیست و نمی توان کنترل پسخورد مناسبی برای پایدار کردن این سیستم پیدا کرد.

مثال ۶-۵ مدل تاخیر زمانی زیر را در نظر بگیرید،

$$\dot{x}(t) = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} -1 & -1 \\ 0 & -0.9 \end{bmatrix} x(t-h) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

قبل از به کار بردن پسخورد، با استفاده از نرم افزار MATLAB، Q_0 و سپس S_0 را به صورت زیر بدست می آوریم،

$$Q_0 = \begin{bmatrix} 1 & 0 \\ -0.0054 & 0.9048 \end{bmatrix}, S_0 = \begin{bmatrix} -1.1183 & -1.1054 \\ 0 & 0.1098 \end{bmatrix}$$

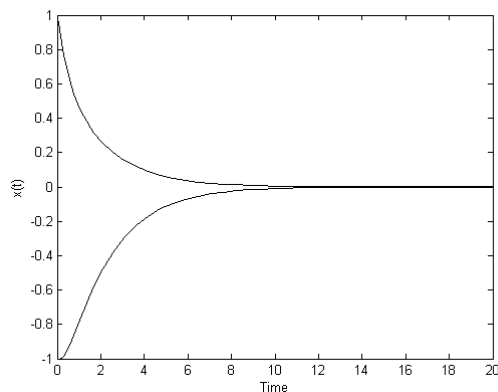
راست ترین مقادیرهای ویژه یا همان مقادیرهای ویژه S_0 ، -1.1183 و 0.1098 هستند و بنابراین زمانی که $h = 0.1s$ می باشد ماتریس S_0 دارای یک قطب ناپایدار است. از آنجایی که سیستم در شرطی که برای کنترل پذیری نقطه ای بیان کردیم صدق می کند، آزمون کنترل پذیری می تواند با استفاده از نرم افزار MATLAB و تابع `pwcontr_test` انجام شود، بنابراین با استفاده از تخصیص مقدار ویژه می

توان یک کنترل کننده پسخورد مناسب برای پایدار کردن این سیستم طراحی کرد.

برای مثال زمانی که قسمت حقیقی راست ترین مقدار ویژه را در موقعیت -0.5 در نظر می گیریم، که به دلخواه انتخاب می شود، بهره های بدست آمده،

$$K = [-0.7453 \quad -1.2219]; K_d = [0.2581 \quad -0.2240]$$

می باشند که با استفاده از نرم افزار MATLAB و تابع `place_dde` بدست می آیند. به این ترتیب با به کار بردن بهره های مورد نظر همانطور که در شکل ۷-۵ نشان داده شده است می توان سیستم را پایدار کرد و راست ترین مقدار ویژه را به موقعیت مورد نظر در صفحه مختلط انتقال داد.



شکل ۷-۵: پاسخ سیستم بعد از به کار بردن پسخورد $K = [-0.7453 \quad -1.2219]; K_d = [0.2581 \quad -0.2240]$

در ادامه این پایان نامه از روش تابع لامبرت به عنوان روشی جدید برای کنترل سیستم های ناوردای زمانی خطی با تاخیر در متغیر حالت و ورودی استفاده می کنیم و نتیجه را با ارائه یک مثال نشان می دهیم.

سیستم های با تاخیر در متغیر حالت و ورودی ۴-۲-۵

سیستم کنترل پذیر از معادلات دیفرانسیل با تاخیر در متغیر حالت و ورودی را در نظر بگیرید،

$$\dot{x}(t) = Ax(t) + A_d x(t-h) + Bu(t) + B_d u(t-h) \quad (۳۴-۵)$$

با در نظر گرفتن $u(t) = Kx(t)$ سیستم حلقه بسته به صورت زیر بدست می آید،

$$\dot{x}(t) = (A + BK)x(t) + (A_d + B_dK)x(t - h) \quad (35-5)$$

بهره K با تخصیص مقادارهای ویژه و با استفاده از الگوریتم زیر تعیین می شود.

الگوریتم ۵-۵

گام ۱- ابتدا مقادارهای ویژه مورد نظر $\lambda_{i,desired}$ را برای $i = 1, \dots, n$ انتخاب می کنیم و قرار می دهیم،

$$\lambda_i(S_0) = \lambda_{i,desired} \quad (36-5)$$

که در آن $\lambda_i(S_0)$ ، i امین مقدار ویژه از S_0 است.

گام ۲- دو ماتریس ضرایب جدید $A = A + BK, A_d = A_d + B_dK$ را در معادله (۳۷-۵) به کار می بریم،

$$W_k(A_d h Q_k) e^{W_k(A_d h Q_k) + Ah} = A_d h \quad (37-5)$$

و این معادله را برای بدست آوردن Q_0 و برای شاخه اصلی حل می کنیم. توجه کنید که K ماتریس مجهول و با همه درایه های نامعلوم می باشند و Q_0 یک تابع از مجهول K می باشد.

گام ۳- ماتریس Q_0 بدست آمده را در معادله (۳۸-۵) قرار می دهیم،

$$S_k = \frac{1}{h} W_k(A_d h Q_k) + A \quad (38-5)$$

سپس S_0 و مقادارهای ویژه آن را به صورت تابعی از ماتریس مجهول K محاسبه می کنیم.

گام ۴- معادله (۳۶-۵) را با استفاده از روش های عددی مانند fsolve در نرم افزار MATLAB حل می کنیم.

به این ترتیب کنترل کننده پسخورد K بدست می آید و سیستم حلقه بسته پایدار می شود.

مثال ۷-۵ سیستم تاخیری زیر را در نظر بگیرید،

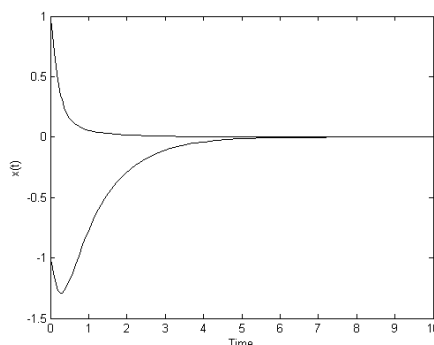
$$\dot{x}(t) = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} -1 & -1 \\ 0 & -0.9 \end{bmatrix} x(t-h) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t-h), u(t) = Kx(t)$$

قبل از به کار بردن پس‌خورده (برای مثال، $K = 0$) راست‌ترین مقدارهای ویژه یا همان مقدارهای ویژه S_0 ، -1.1183 و 0.1098 هستند و بنابراین زمانی که $h = 0.1s$ می‌باشد سیستم کنترل پذیر، ناپایدار است. بنابراین با استفاده از تخصیص مقدار ویژه می‌توان یک کنترل کننده پس‌خورده مناسب برای پایدار کردن سیستم کنترل پذیر، طراحی کرد و با انتخاب پس‌خورده K مقدارهای ویژه را به موقعیت‌های مورد نظر در صفحه مختلط انتقال داد.

در این مثال نیز هدف پیدا کردن $K = [k_1 \quad k_2]$ به گونه‌ای است که راست‌ترین مقدار ویژه در موقعیت -1 قرار بگیرد که با استفاده از نرم‌افزار Matlab و تابع `place_dde` ماتریس پس‌خورده حالت را به صورت زیر بدست می‌آوریم،

$$K = [-0.0760 \quad -1.6548] \quad (39-5)$$

و به این ترتیب سیستم حلقه بسته پایدار می‌شود. شکل ۸-۵ نیز پاسخ سیستم را بعد از به کار بردن پس‌خورده بدست آمده نشان می‌دهد.



شکل ۸-۵: پاسخ سیستم بعد از به کار بردن پس‌خورده $K = [-0.0760 \quad -1.6548]$

همچنین می‌توان از روش تابع لامبرت برای پایدارسازی سیستم کنترل پذیر خطی زیر،

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (40-5)$$

استفاده کرد. به این صورت که با در نظر گرفتن $u(t) = K_d x(t-h)$ سیستم حلقه بسته به صورت

زیر بدست می آید،

$$\dot{x}(t) = Ax(t) + BKx(t - h) \quad (۴۱-۵)$$

سپس با در نظر گرفتن ماتریس های ضرایب جدید و مشابه حالت های قبل، ماتریس پسخورد حالت مناسب را بدست می آوریم و به این ترتیب سیستم حلقه بسته پایدار می شود.

فصل ششم

خلاصه و نتایج

۱-۶ خلاصه

در این پایان نامه، روشی جدید برای آنالیز و کنترل سیستم های تاخیری که در سال های اخیر مطرح شده را معرفی کردیم. جواب های تحلیلی برای سیستم هایی از معادلات دیفرانسیل تاخیری بدست آمد. همچنین نتایج جدیدی برای طراحی کنترل کننده پسخورد و برای یک رده از سیستم های تاخیری ارائه شد. به طوریکه برای یک سیستم داده شده که به وسیله معادلات دیفرانسیل تاخیری نشان داده می شود براساس تابع لامبرت پایداری تعیین می شود و اگر سیستم ناپایدار باشد بعد از بررسی کنترل پذیری سیستم، به وسیله تخصیص مقدار ویژه یک پسخورد پایدار کننده طراحی می شود و سرانجام سیستم حلقه بسته از این معادلات می تواند پایدار شود. مثال های عددی نیز برای توضیح این روش ارائه شد. از آنجایی که معادلات دیفرانسیل تاخیری دارای یک طیف نامتناهی هستند و امکان تخصیص برای همه مقدارهای ویژه سیستم حلقه بسته وجود ندارد به همین دلیل سعی شد مقدار های مورد نظر، به یک زیرمجموعه از مقدارهای ویژه که در تعیین پایداری سیستم نقش مهمی دارند، اختصاص داده شود.

۲-۶ نتایج

در این پایان نامه نتایج جدیدی برای کنترل و پایدارسازی سیستم های تاخیری ارائه گردید به طوریکه با استفاده از تابع لامبرت و با اختصاص مقادیر ویژه مناسب به سیستم حلقه بسته، کنترل کننده پسخورد را به گونه ای محاسبه کردیم که بتوان سیستم تاخیری ناپایدار را به صورت پایدار، کنترل نمود. ما این روش را با روش های دیگر مقایسه کردیم (بخش ۲-۵ را ببینید) تعدادی از آنها فاقد عمومیت هستند و ممکن است برای مسائل خاصی با شکست مواجه شوند. روش مطرح شده در [۱۶]

که یک روش تکراری براساس حساسیت مقدرهای ویژه نسبت به بهره کنترل می باشد از سایر روش ها مؤثرتر است. همچنین روش ارائه شده در این تحقیق نیز دارای فوایدی است که از جمله می توان به دقت آن اشاره کرد به این خاطر که عبارت نمایی در معادله مشخصه سیستم در این روش تقریب زده نمی شود بنابراین جوابی که بدست می آید دقیق تر است. از دیگر فواید این روش آن است که فرم جواب بدست آمده مشابه جواب برای معادلات دیفرانسیل معمولی می باشد بنابراین مفهوم ماتریس انتقال حالت در این معادلات را می توان با استفاده از مفهوم ماتریس تابع لامبرت برای معادلات دیفرانسیل تاخیری تعمیم داد در نتیجه برخی روش ها برای آنالیز و کنترل سیستم هایی از معادلات دیفرانسیل معمولی شامل آنالیز پایداری، کنترل پذیری و روش های تخصیص مقدار ویژه برای طراحی کنترل کننده پسخورد خطی را می توان برای سیستم هایی از معادلات دیفرانسیل تاخیری تعمیم داد.

بنابراین تحقیق ارائه شده در این پایان نامه یک روش مؤثر و قابل اجرا برای آنالیز و کنترل سیستم های تاخیری نشان می دهد.

۳-۶ مسئله های باز و کارهای آینده

اولین مسأله ای که می توان به آن اشاره کرد این است که از آنجایی که روش استفاده از ماتریس تابع لامبرت وابسته به تعیین ماتریس Q_k است و همانطور که قبلاً بیان کردیم برای بدست آوردن Q_k نیز معادله،

$$W_k(A_d h Q_k) e^{W_k(A_d h Q_k) + A h} = A_d h \quad (۱,۶)$$

را برای شرایط اولیه متنوعی به روش عددی و با استفاده از تابع fsolve در نرم افزار MATLAB و برای هر شاخه حل می کنیم اما هنوز شرایطی برای وجود و یکتایی Q_k وجود ندارد.

مسأله دیگر این است که در همه مثال هایی که مطرح شد زمانی که A_d مقدرهای ویژه چندگانه صفر

را نداشته باشد پایداری با استفاده از شاخه اصلی ($k = 0$) از تابع لامبرت تعیین می شد که این مسأله برای حالت اسکالر اثبات شده است اما برای برخی فرم های خاص از حالت ماتریسی هنوز اثبات کلی وجود ندارد.

همچنین رابطه بین کنترل پذیری و تخصیص مقدار ویژه برای سیستم هایی از معادلات دیفرانسیل تاخیری یک مسأله باز دیگر است. اگر چه کنترل کننده های پسخورد می توانند برای سیستم های تاخیری طراحی شوند اما به این معنا نیست که راست ترین مقدار ویژه به دلخواه اختصاص داده می شود زیرا با توجه به پارامترها و ساختار سیستم، محدودیت هایی وجود دارد و در بدترین حالت، برخی از سیستم ها ممکن است با هر مقدار پسخورد، پایدار نشوند.

۴-۶ پیشنهادات

در ادامه می توان از روش تابع لامبرت با محاسبه شعاع پایداری سیستم برای طراحی کنترل کننده مقاوم^۱ استفاده کرد. همچنین تاکنون در اکثر مسائل حل شده از این فرض استفاده کردیم که تمام حالت ها قابل اندازه گیری هستند در حالیکه در برخی کاربردهای مهندسی به خاطر هزینه و مشکلات در پیاده سازی حسگر، بعضی حالت ها به طور مستقیم قابل اندازه گیری نیستند برای همین تشخیص یک برآوردکننده حالت و یا مشاهده کننده^۲ لازم می باشد. بنابراین می توان این روش را با تخمین متغیر های حالت و با استفاده از تخصیص مقدار ویژه برای طراحی پسخورد مشاهده کننده به کار برد. همچنین از روش تابع لامبرت برای تحلیل حساسیت نیز استفاده می شود که می توان به کاربرد آن در آنالیز حساسیت برای مدل بیماری HIV با یک تاخیر درون سلولی اشاره کرد به طوریکه عبارت تحلیلی برای حساسیت راست ترین مقدار ویژه، با گرفتن مشتق از معادله مشخصه سیستم نسبت به هر یک از پارامترها بدست می آید. به این ترتیب با قراردادن مقدار هر کدام از پارامترها و راست ترین

^۱Robust Controller

^۲Observer

مقدار ویژه در روابط بدست آمده، با استفاده از علامت مشتق تعیین می کنیم آیا افزایش مقدار یک پارامتر، راست ترین مقدار ویژه را افزایش یا کاهش خواهد داد؟ بنابراین اگر حساسیت نسبت به یک پارامتر مثبت باشد نتیجه می گیریم که افزایش مقدار یک پارامتر باعث می شود راست ترین مقدارهای ویژه به سمت راست صفحه مختلط انتقال پیدا کنند و سیستم ایمنی بدن بیشتر ناپایدار شود و به عکس. به این ترتیب با تغییر مقدار پارامترها سعی می کنیم سیستم پایدار شود.

ضمیمه A

کد MATLAB

در این قسمت کدهای متلب استفاده شده در این پایان نامه را می آوریم.

```
% DDE_Sol = find_Sk(A,Ad,h,N,Q_initial)
% The argument N and Q_initial are optional
% Find Sk for each branch given A, Ad and h for a DDE system
% x_dot(t) = A*x(t) + Ad*x(t-h)
% Q_initial is the initial condition for Qk
% N is the index of the branches of interest;
function DDE_Sol = find_Sk(A,Ad,h,N,Q_initial,thredhold)
options_def = optimset('Display','iter');
if nargin < 3
    error('No enough input arguments')
elseif nargin < 4
    N = 0; % default is to solve for principal branch
    Q_initial = expm(-A*h);
    options = options_def;
elseif nargin < 5
    Q_initial = expm(-A*h);
    options = options_def;
elseif nargin < 6
    options = options_def;
else
    if thredhold~=0 % if the thredhold is specified
        options = options_def;
        options.TolFun = thredhold;
    else
        options = options_def;
    end
end
dim_Ad = length(Ad); % system order
DDE_Sol = {};
if dim_Ad == 1 % scalar case
    if Ad == 0
        warning('First order ODE; x = exp(A*t)*x_0')
    else
        for i = 1: length(N)
            DDE_Sol{i}.Sk = lambertw(N(i),Ad*h*exp(-A*h))/h+A;
            DDE_Sol{i}.branchID = N(i);
            error1 = norm(lambertw(N(i),Ad*h*exp(-A*h))-Ad*h-h*DDE_Sol{i}.Sk);
            error2 = norm(DDE_Sol{i}.Sk*eye(length(A))-A-Ad*expm(-DDE_Sol{i}.Sk*h));
            DDE_Sol{i}.error = [error1 error2];
        end
    end
else % matrix case
    for i = 1: length(N)
        [DDE_Sol{i}.Sk,DDE_Sol{i}.Q,error1,error2]=QS_matrix(A,Ad,h,N(i),Q_initial,options);
        DDE_Sol{i}.branchID = N(i);
        DDE_Sol{i}.error = [error1 error2];
    end
end
```

```

end
% currentQ = DDE_Sol{i}.Q % display the Q for current iteration
% find Sk and Qk for matrix case
function [S,Q1,error1,error2] = QS_matrix(A,Ad,h,branch_id,Q_initial,options)
if sum(sum(Q_initial == inf))>0
    reduced = 1; % Reduced elements for Q
    notfixed = Q_initial ~= inf; % find elements not fixed
    notfixed = reshape(notfixed,1,[]);
    res_Q = reshape(Q_initial,1,[]);
    reduced_Q = res_Q(notfixed); % formulate the new vector of design variables
    Q = fsolve(@LambQ,reduced_Q,options,A,Ad,h,branch_id,reduced,notfixed);
else
    reduced = 0; % full elements for Q
    fixed = 0;
    Q = fsolve(@LambQ,Q_initial,options,A,Ad,h,branch_id,reduced,fixed);
end
if reduced ==0
    Q1 = Q;
else
    % if the reduced Q is used, convert the results back to matrix with
    % original dimension
    Q1 = zeros(1,length(A)^2);
    position = find(notfixed==1);
    for k=1:length(position)
        Q1(position(k)) = Q(k);
    end
    Q1 = reshape(Q1,length(A),[]);
end
H = Ad*h*Q1;
W = lambertw_matrix(branch_id,H);
S = W/h+A;
error1 = norm(W*expm(W+A*h)-Ad*h); % must be the solution to the lambert w function eqn.
error2 = norm(S*eye(length(A))-A-Ad*expm(-S*h)); % must be the solution to the char. eqn.
error = [error1 error2]; % both must be close to zero
-----
% [CI, uniqueness]= find_CI(S,A,Ad,h)
% Find CI_k for the kth branch given A, Ad, h and Sk for this branch
% Important assumption: Any two eigenvalues from different branches are
% distinct !
% When Ad is invertible, S should be a matrix; when Ad is singular, S
% should be a scalar
% If the solution is unique, uniqueness=1; otherwise uniqueness=0
function [CI, uniqueness]= find_CI(A,Ad,h,S,g0,x0)
sys_order = length(A);
% scalar case
if sys_order==1
    CI=(x0+Ad*preshape_conv(g0,S,h))/(1+Ad*h*exp(-S*h)); % (x0+Ad*G(Sk))/(1+Ad*h*exp(-S*h))
    uniqueness = 1;
    return
end
% matrix case
if length(S)==sys_order % general case, S is a matrix
    E = eig(S);
    syms s
    L1_num = 1;
    for i=1:sys_order
        L1_num = L1_num*(s-E(i));
    end
    char_eqn = det(s*eye(sys_order)-A-Ad*expm(-s*h));

```

```

L1 = diff(L1_num)/diff(char_eqn);
R_matrix = [];
L_matrix = [];
for k=1:sys_order
    R1_num = adjugate(eye(sys_order)*E(k) - S);
    L1_num = subs(L1,s,E(k));
    L2_num = adjugate(E(k)*eye(sys_order)-A-Ad*exp(-E(k)*h));
    L3_num = x0+Ad*preshape_conv(g0,E(k),h); % (x0+Ad*G(Sk))
    R_matrix = [R_matrix;R1_num]; % stacked right side
    L_matrix = [L_matrix;L1_num*L2_num*L3_num]; % stacked left side
end
uniqueness = rank(L_matrix)==sys_order;
% for having an unique solution, matrix should have full column rank
CI = pinv(R_matrix)*L_matrix; % use generalized pseudo inverse to calculate the solution
elseif sys_order>1 && length(S)==1 % hybrid branch case, S is a scalar
    syms s
    char_eqn = det(s*eye(sys_order)-A-Ad*expm(-s*h));
    L1 = 1/diff(char_eqn);
    L1_num = subs(L1,s,S);
    L2_num = adjugate(S*eye(sys_order)-A-Ad*exp(-S*h));
    L3_num = x0+Ad*preshape_conv(g0,S,h);
    CI = L1_num*L2_num*L3_num;
    uniqueness = (L1_num~=0);
else
    error('Dimension of A and S are not compatible');
-----
% [CN, uniqueness]= find_CN(S,A,Ad,h)
% Find CN_k for the kth branch given A, Ad, h and Sk for this branch
% Important assumption: Any two eigenvalues from different branches are distinct !
% When Ad is invertible, S should be a matrix; when Ad is singular, S should be a scalar
% If the solution is unique, uniqueness=1; otherwise uniqueness=0
function [CN, uniqueness]= find_CN(A,Ad,h,S)
E = eig(S);
sys_order = length(A);
% scalar case
if sys_order == 1
    CN = 1/(1+Ad*h*exp(-S*h));
    uniqueness = 1;
    return
end
% matrix case
if length(S) == sys_order % general case, S is a matrix
    syms s
    L1_num = 1;
    for i=1:sys_order
        L1_num = L1_num*(s-E(i));
    end
    char_eqn = det(s*eye(sys_order)-A-Ad*expm(-s*h)); % char eqn.
    L1 = diff(L1_num)/diff(char_eqn);
    R_matrix = [];
    L_matrix = [];
    for k=1:sys_order
        R1_num = adjugate(eye(sys_order)*E(k) - S);
        L1_num = subs(L1,s,E(k));
        L2_num = adjugate(E(k)*eye(sys_order)-A-Ad*exp(-E(k)*h));
        R_matrix = [R_matrix;R1_num]; % stacked right side
        L_matrix = [L_matrix;L1_num*L2_num]; % stacked left side
    end
    uniqueness = rank(L_matrix)==sys_order;

```

```

% for having an unique solution, matrix should have full column rank
CN = pinv(R_matrix)*L_matrix; % use generalized pseudo inverse to calculate the solution
elseif sys_order>1 && length(S)==1 % hybrid branch case, S is a scalar
syms s
char_eqn = det(s*eye(sys_order)-A-Ad*expm(-s*h));
L1 = 1/diff(char_eqn);
L1_num = subs(L1,s,S);
L2_num = adjugate(S*eye(sys_order)-A-Ad*exp(-S*h));
CN = L1_num*L2_num;
uniqueness = (L1_num~=0);
else
error('Dimension of A and S are not compatible');
end

```

```

% calculate matrix lambert w function
function W = lambertw_matrix(K,X)
if nargin == 1 % Only calculate principal branches
X = K;
K = 0;
end
rank_X = rank(X);
dim_X = length(X); % system order
LMW = zeros(dim_X);
[V,D] = jordan(X); % transform to Jordan form
% check superdiagonal terms
super_diag = [];
if dim_X==1;
super_diag = 0;
else
for i = 2:dim_X
super_diag = [super_diag D(i-1,i)];
end
end
% calculate the Lambert W func. for diagnal terms
for i = 1:dim_X
if D(i,i)==0
% use principal branch for 0 eigenvalue
LMW(i,i)=lambertw(0,D(i,i)); % warning('Ad is singular, hybrid branch is used')
else
LMW(i,i)=lambertw(K,D(i,i));
end
end
% calculate the Lambert W func. for other upper triangle terms
if sum(super_diag)==0
flag =0; % all jordan blocks have a size of 1
else
flag =1; % some jordan blocks have a size > 1
block_start = 1; % where an jordan block begins
block_end = 1; % where an jordan block ends
for j=2:dim_X
if j>=block_end
if super_diag(j-1)==1
block_start = j-1;
if ~isempty(find(super_diag(j-1:end)==0))
block_end = min(find(super_diag(j-1:end)==0))+j-2;
else
block_end = dim_X;
end
end
for k1 =block_start:block_end

```

```

        for k2 = k1: block_end
            if D(block_start,block_start)~=0
                LMW(k1,k2)=der_lambertw(K,D(block_start,block_start),k2-k1)/factorial(k2-k1);
                % (k2-k1)th derivate of lambertw(D(i))/(k2-k1)!
            end
        end
    end
end
end
end
end
end
W = V*LMW*V^-1; % similarity transformation
-----
% Piecewise controllability test for a system of DDEs
%  $\dot{x}(t) = A*x(t) + Ad*x(t-T)+B*u(t)$ 
% Check the linear independence of the rows of  $(sI-A-Ad*exp(-s*h))^{-1}*B$ 
% Let  $[C_1(s);C_2(s); \dots ;C_n(s)]= (sI-A-Ad*exp(-s*h))B$ ; If there exist  $p_1, p_2, \dots, p_n$ , such that
%  $[C_1(p_1) C_1(p_2) \dots C_1(p_n); \dots; C_n(p_1) C_n(p_2) \dots C_n(p_n)]$  has full
% rank, then  $C_1(s), C_2(s), \dots, C_n(s)$  are linearly independent
function contr_flag = pwcontr_test(A,Ad,B,h)
thred = 1e-5;
if nargin < 4
    error('No enough input arguments')
end
dim = length(A); % dimension of the system
syms s
char = s*eye(dim)-A-Ad*exp(-s*h);
contr_gram = char^-1*B;
runs = 10; % test 10 times. If failed everytime, the system is NOT piecewise controllable; if passed for a
run, the system is piecewise controllable
contr_flag = 0; % default is set to be not piecewise controllable
for j = 1:runs
    p = 10*rand(dim,1); % random variable [p1 p2 p3 ... pn]
    contr_matrix = [];
    for i = 1:dim
        contr_matrix = [contr_matrix subs(contr_gram, p(i))];
    end
    % formulating the matrix [C(p1) C(p2) .... C(pn)]
    if det(contr_matrix) > thred
        contr_flag = 1;
        disp('The system of DDEs is piecewise controllable.')
        break;
    end
end
end
if contr_flag==0
    disp('The system of DDEs is NOT piecewise controllable.')
end
-----
% Find a feedback gain K to place the rightmost eigenvalue of the closed-loop at desired place
% contr mode 1:  $u = -K*x(t)$ 
% the closed-loop system is  $\dot{x}(t) = (A-B*K)*x(t) + Ad*x(t-h)$ 
% contr mode 2:  $u = -Kd*x(t-h)$ 
% the closed-loop system is  $\dot{x}(t) = A*x(t) + (Ad-B*Kd)*x(t-h)$ 
% contr mode 3:  $u = -K*x(t) - Kd*x(t-h)$ 
% the closed-loop system is  $\dot{x}(t) = (A-B*K)*x(t) + (Ad-B*Kd)*x(t-h)$ 
% observer mode 4: observer gain is L
% the closed-loop system is  $\dot{x}(t) = (A-L*C)*x(t) + Ad*x(t-h)$ 
% default mode is 1.
%  $K = \text{place\_dde}(A,Ad,B,h,branch,pole\_desired,contr\_mode,K0,Q\_ini,thredhold)$ 

```

```

% The arguments contr_mode, K0, Q_ini and threshold are optional
% pole_desired must be a scalar. For complex values, e.g., you can use either -1+i or -1-i.
function K = place_dde(A,Ad,B,h,branch,pole_desired,contr_mode,K0,Q_ini,threshold)
options_K_def = optimset('Display','final');
th2 = 0;
[B_row, B_col] = size(B);
if nargin <=5
    warning('No enough input arguments.')
elseif nargin ==6
    contr_mode = 1;% default mode 1:u(t) = -Kx(t)
    K0 = ones(B_col,length(A));
    Q_ini = expm(-A*h); % use e^(-Ah) to initialize Q
elseif nargin ==7 % only contr mode is selected
    if contr_mode <=2
        K0 = ones(B_col,length(A)); Q_ini = expm(-A*h); % mode 1 and 2
    elseif contr_mode ==3
        K0 = [ones(B_col,length(A)) zeros(B_col,length(A))]; Q_ini = expm(-A*h); % mode 3
    elseif contr_mode ==4
        K0 = ones(length(A),B_row); Q_ini = expm(-A*h); % observer mode
    else
        warning('mode must be <= 4');
    end
elseif nargin ==8
    Q_ini = expm(-A*h);
elseif nargin ==10
    if threshold(1)~=0
        options_K_def.TolFun = threshold(1);
    end
    if threshold(2)~=0
        th2 = threshold(2); % if threshold is not zero, use it to replace the default threshold
    end
end

if length(pole_desired)>1
    warning('Pole_desired is a scalar, please enter either a real number, or a complex number');
end
K = fsolve(@find_K,K0,options_K_def,A,Ad,B,h,branch,pole_desired,contr_mode,Q_ini,th2)

function rtn=find_K(X,A,Ad,B,h,branch,pole_desired,contr_mode,Q_initial,threshold2)
[B_row, B_col] = size(B);
% calculate the closed-loop system
if contr_mode == 1
    K = X; Kd = zeros(B_col,length(A));A_cl = A-B*K;Ad_cl = Ad-B*Kd;
elseif contr_mode ==2
    Kd = X; K = zeros(B_col,length(A));A_cl = A-B*K;Ad_cl = Ad-B*Kd;
elseif contr_mode ==3
    K = X(:,1:length(A)); Kd = X(:,length(A)+1:end);A_cl = A-B*K;Ad_cl = Ad-B*Kd;
elseif contr_mode ==4
    A_cl = A-X*B;Ad_cl = Ad;
end
% can do the assignment for only one branch!
if length(branch)~=1
    error('Please select one and only one branch');
end
% find the solution for the closed-loop system
DDE_Sol = find_Sk(A_cl,Ad_cl,h,branch,Q_initial,threshold2);
% showing the current result
eig_S = eig(DDE_Sol{1}.Sk)
% calculate the errors of eigenvalues

```

```

if isreal(pole_desired)
    rtn = (max(real(eig_S))-pole_desired)^2; % match only the real part
else
    real_eig_diff = max(real(eig_S))-real(pole_desired);
    imag_eig_diff = max(imag(eig_S(find(real(eig_S)==max(real(eig_S))))))-abs(imag(pole_desired));
    rtn = [real_eig_diff^2;imag_eig_diff^2]; % match both real and imag parts
end

-----

% find Sk and CI
A = -0.1; Ad = -0.1; h=1;
DDE_Sol = find_Sk(A,Ad,h,0);
Sk = DDE_Sol{1}.Sk;
Sk = eig(DDE_Sol{1}.Sk)
g = [0];x0 = [10];
[CI,uniqueness] = find_CI(A,Ad,h,Sk,g,x0)

-----

% Matrix lambert W function
W = [1 1 1;0 1 2;0 0 3]; branch = -1; % repeated eigenvalue
jordan(W)
lambertw_matrix(branch,W)

-----

% Matrix Function: Compute Qk
global h k1 k2 A Ad I % Define variable for fsolve
k1 = 0;k2=k1; % Branch of the matrix Lambert W function
I = eye(2); % for identity matrix
h = 0.1; % time-delay
A = [-1 -3;2 -5]; Ad = [1.66 -0.697;0.93 -0.330]; % coefficient matrices of the system A=-Ad;B=-A
x0=I; % initial condition for iteration => identity matrix
tol=1e-8;options = optimset('MaxFunEvals',1000,'TolFun',1e-8); % iteration options
X = fsolve(@equations,x0,options)

function rtn=equations(x)
global h k1 k2 A Ad I
X = [x(1) x(2) x(3) x(4)];
Q = [x(1) x(2);x(3) x(4)]; D = Ad*h*Q;
[v,d] = eig(D);
W = v*[lambertw(k1,d(1,1)) 0;0 lambertw(k2,d(2,2))]*inv(v); % for matrix Lambert W function
EX = expm(W+A*h);
Left = (W*EX);
Right = Ad*h;
rtn = [Left(1,1)-Right(1,1)
        Left(1,2)-Right(1,2)
        Left(2,1)-Right(2,1)
        Left(2,2)-Right(2,2)];

-----

% find Sk
B = -A
Q = [X(1) X(2);X(3) X(4)] % result for Qk
D = Ad*h*Q; [v,d] = eig(D); % To compute Sk
W = v*[lambertw(k1,d(1,1)) 0;0 lambertw(k2,d(2,2))]*inv(v);
Sk = 1/h*W-B % result for Sk

-----

% find CI
h = 1;
A = [-1 -3;2 -5]; Ad = [1.66 -0.697;0.93 -0.330];
% coefficient matrices of the system A=-Ad;B=-A
Sk = [0.3055 -1.4150;2.1317 -3.3015];
g = [1;0];x0 = [1;0];
[CI,flag] = find_CI(A,Ad,h,Sk,g,x0)

```

```

-----
% find Sk,CI and CN
A = -1; Ad = -1; h=1;
DDE_Sol = find_Sk(A,Ad,h,1);
Sk = DDE_Sol{1}.Sk;
Sk = eig(DDE_Sol{1}.Sk)
[CN,uniqueness] = find_CN(A,Ad,h,Sk)
g = [1];x0=[1];
[CI,uniqueness] = find_CI(A,Ad,h,Sk,g,x0)
-----
% find CI and CN
h = 1;
A = [-1 -3;2 -5]; Ad = [1.66 -0.697;0.93 -0.330]; % coefficient matrices of the system A=-Ad;B=-A
Sk = [-0.3499-4.9801i -1.6253+0.1459i;2.4174+0.1308i -5.1048-4.5592i];
[CN,flag] = find_CN(A,Ad,h,Sk)
g = [1;0]; x0=[1;0];
[CI,flag] = find_CI(A,Ad,h,Sk,g,x0)
-----
% find Sk
A = [0 1;-4.6985 0]; Ad = [0 0;0 0.5]; h=1;
Q_ini =expm(-A*h);
DDE_Sol = find_Sk(A,Ad,h,0,Q_ini);
DDE_Sol{1}.Sk
Sk = eig(DDE_Sol{1}.Sk)
-----
% compute ad
function num1
a = -1;
lambda_des = -0.5; % place the right most pole at -0.5
ad_ini = 0;
h = 1;
options=optimset('Display','iter'); % Option to display output
ad = fsolve(@assign_lambert,ad_ini,options,a,h,lambda_des)
display(['right most pole is placed at: num2str(lambertw(0,ad*h*exp(-a*h))/h+a)])

function F = assign_lambert(ad,a,h,lambda_des)
F = real(lambertw(0,ad*h*exp(-a*h))/h+a)-lambda_des;
-----
% compute k
function num2
a = 1;
ad = -1;
lambda_des = -1; % place the right most pole at -1
k_ini = 2;
h = 1;
options=optimset('Display','iter'); % Option to display output
k = fsolve(@assign_lambert,k_ini,options,a,ad,h,lambda_des)

function F = assign_lambert(k,a,ad,h,lambda_des)
F = real(lambertw(0,ad*h*exp(-(a+k)*h))/h+(a+k))-lambda_des;
-----
% compute k
function num3
a = -0.1;
ad_res = 0.1;
lambda_des = -0.5; % place the right most pole at -0.5
k_ini = 0;
h = 1;
options=optimset('Display','iter'); % Option to display output

```



```
K = fsolve(@assign_lambert,k_ini,options,a,ad_res,h,lambda_des)
display(['right most pole is placed at:' num2str(lambertw(0,ad_res*K_placed*h*exp(-a*h))/h+a)])
```

```
function F = assign_lambert(k,a,ad_res,h,lambda_des)
F = real(lambertw(0,ad_res*k*h*exp(-a*h))/h+a)-lambda_des;
```

```
-----
% compute K
h = 0.2;% time-delay
A = [0 1;-1 0.1] ; Ad = [0 0 ;0 0] ; B = [0;1]; % open loop
% pole_desired=[-1+2.5i];
pole_desired=[-1+2i];
% pole_desired=[-1+1.5i];
Q_ini = [inf inf;1 1];
Kd_ini = [1 1];
contr_mode = 2;
branch =0;
% K = place_dde(A,Ad,B,h,branch,pole_desired,contr_mode);
threshold = [1e-8, 1e-8];
K = place_dde(A,Ad,B,h,branch,pole_desired,contr_mode);
```

```
-----
% compute K
h = 0.2;% time-delay
A = [0 1;-4.6985 0] ; Ad = [0 0 ;0 0] ; B = [0;0.25]; % open loop
% pole_desired=[-1+2.5i];
pole_desired=[-1+2i];
% pole_desired=[-1+1.5i];
Q_ini = [inf inf;1 1];
Kd_ini = [1 1];
contr_mode = 2;
branch =0;
% K = place_dde(A,Ad,B,h,branch,pole_desired,contr_mode);
threshold = [1e-8, 1e-8];
K = place_dde(A,Ad,B,h,branch,pole_desired,contr_mode);
```

```
-----
% contr.test
h = 1; % time-delay
A = [0 0;0 1]; Ad = [-1 -1;0 -0.9]; h = 1; B = [0;1];
pwcontr_test(A,Ad,B,h)
```

```
-----
% find Sk
A = [0 0;0 1]; Ad = [-1 -1;0 -0.9]; h=0.1;
Q_ini =expm(-A*h);
DDE_Sol = find_Sk(A,Ad,h,0,Q_ini);
DDE_Sol{1}.Sk
Sk = eig(DDE_Sol{1}.Sk)
```

```
-----
% compute K and Kd
h = 0.1; % time-delay
A = [0 0;0 1] ; Ad = [-1 -1;0 -0.9] ; B = [0;1]; % open loop
pole_desired=[-1];
Q_ini = expm(-A*h);
Kd_ini = [1 1];
K_ini=[1 1]
contr_mode = 3;
branch =0;
% K = place_dde(A,Ad,B,h,branch,pole_desired,contr_mode);
threshold = [1e-8, 1e-8];
K = place_dde(A,Ad,B,h,branch,pole_desired,contr_mode);
```

ضمیمه B

ضمیمه برای فصل ۵

1.B محدودیت در تخصیص مقدار ویژه

همانطور که قبلاً نیز بیان کردیم، براساس روش تابع لامبرت می توان راست ترین قطب سیستم های تاخیری را به موقعیت های مورد نظر در صفحه مختلط انتقال داد و از این روش برای طراحی کنترل کننده پسخورد استفاده کرد.

سیستم زیر را در نظر بگیرید،

$$\dot{x}(t) = ax(t) + u(t-h), a > 0 \quad (1.B)$$

با کنترل پسخورد خطی،

$$u(t) = fx(t) \quad (2.B)$$

که در آن f پسخورد حالت است. در چنین حالتی سیستم حلقه بسته به صورت،

$$\dot{x}(t) = ax(t) + fx(t-h) \quad (3.B)$$

بدست می آید. راست ترین ریشه از معادله مشخصه سیستم (3.B) به صورت زیر بدست می آید،

$$s_0 = \frac{1}{h} W_0(fhe^{-ah}) + a \quad (4.B)$$

هدف اصلی از انتخاب بهره f پایدار کردن سیستم است به طوریکه می توان با قرار دادن راست ترین مقدار ویژه در موقعیت مورد نظر و با حل معادله،

$$\lambda_{desired} = \frac{1}{h} W_0(fhe^{-ah}) + a \quad (5.C)$$

برای بدست آوردن بهره مناسب f ، سیستم را پایدار کرد.

مثال ۱ فرض کنید $a = 1, h = 0.1s$ و راست ترین مقدار ویژه مورد نظر، $\lambda_{desired}$ ، -1 باشد آنگاه

معادله زیر را برای بدست آوردن بهره f حل می کنیم،

$$-1 = \frac{1}{0.1} W_0(f \times 0.1 \times e^{-0.1}) + 1$$

و بدست می آوریم $f = -1.8097$.

اگر چه یک کنترل کننده پسخورد می تواند برای سیستم های تاخیری طراحی شود اما نتیجه بالا به این معنا نیست که به راست ترین مقدار ویژه می توان هر مقداری را اختصاص داد. با توجه به پارامترهای سیستم مخصوصاً تاخیر زمانی، محدودیت هایی در تخصیص مقدار ویژه وجود دارد و در بدترین حالت برخی سیستم ها نمی توانند با هر مقدار پسخورد f پایدار شوند.

همانطور که می دانیم هر شاخه $W_k(H)$ از تابع لامبرت که H یک مقدار مختلط است، دارای تصویر مخصوص به خود می باشد همچنین تصویر هر شاخه نسبت به محور حقیقی متقارن است و قسمت حقیقی شاخه اصلی W_0 دارای مینیمم مقدار -1 در $H = \frac{-1}{e}$ است. بنابراین قسمت حقیقی شاخه اصلی همیشه بزرگتر یا مساوی -1 است که نامساوی زیر را در خصوص حد پایین در تخصیص راست ترین ریشه نتیجه می دهد،

$$Re\{s_0\} = Re\{\lambda_{desired}\} = \frac{1}{h} \underbrace{Re\{W_0(fhe^{-ah})\}}_{\geq -1} + a \geq \frac{-1}{h} + a \quad (6.B)$$

در این نامساوی عکس تاخیر زمانی مانند یک عامل وزنی عمل می کند. اگر h کوچک باشد عبارت پایداری $W_0(fhe^{-ah})$ که می تواند با پسخورد f تنظیم شود دارای اثر نسبتاً بیشتری روی s_0 است و به عکس. بنابراین زمانی که سیستم ناپایدار است می توان گفت که اگر تاخیر زمانی بزرگ باشد، پایدار کردن سیستم مشکل تر خواهد بود. بنابراین با فرض اینکه پسخورد f می تواند هر مقدار حقیقی باشد حتی اگر عبارت $W_0(fhe^{-ah})$ با انتخاب مقدار f به صورت زیر کاهش پیدا کند،

$$fhe^{-ah} = -\frac{1}{e} \rightarrow f = -\frac{1}{he^{1-ah}} \quad (7.B)$$

آنگاه یک مقدار h^* وجود دارد که امکان پایداری برای تاخیر زمانی h ، بزرگتر از h^* وجود ندارد. یعنی،

$$0 = \frac{-1}{h^*} + a \rightarrow h^* = \frac{1}{a} \quad (8.B)$$

برای مثال اگر $a = 1$ آنگاه برای هر $h > h^* = 1s$ سیستم نمی تواند با هر مقدار پسخورد f پایدار شود. برای مثال زمانی که $h = 2s$ باشد، راست ترین مقدار ویژه، طبق نامساوی همیشه بزرگتر از $+0.5$ می باشد. به این ترتیب می توان تاخیر زمانی بحرانی برای پایداری سیستم را پیدا کرد. برای مثال اگر $f > 0$ باشد آنگاه آرگومان W_0 همیشه بزرگتر از صفر است بنابراین سیستم هیچگاه پایدار نمی شود.

همچنین با توجه به این که تاخیر زمانی در ورودی، حالت و یا هر دو باشد حالت های متفاوتی از این مسأله وجود دارد. برای مثال حالت زیر را در نظر بگیرید،

$$\dot{x}(t) = ax(t) + a_d x(t-h) + u(t) \quad (9.B)$$

که در آن،

$$u(t) = fx(t) + f_d x(t-h) \quad (10.B)$$

بنابراین راست ترین ریشه به صورت،

$$s_0 = \frac{1}{h} W_0((a_d + f_d)he^{-(a+f)h}) + a + f \quad (11.C)$$

می باشد که در مورد مسائل پایداری و محدودیت های آن می توان مشابه قبل بحث کرد.

- [1] F. M. Asl and A. G. Ulsoy, "Analysis of a system of linear delay differential equations," *ASME Journal of Dynamics Systems, Measurement and Control* 125(2), pp.215-223, 2003.
- [2] Y. Q. Chen and K. L. Moore, "Analytical stability bound for delayed second-order systems with repeating poles using lambert W function," *Automatica*, 38(5), pp.891-895, 2002.
- [3] C. T. Chen, "Linear system theory and design," *Holt, Rinehart and Winston*, New York, 1984.
- [4] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey and D. E. Knuth, "On the lambert W function," *Advances in Computational Mathematics* 5(4), pp.329-359, 1996.
- [5] B. N. Datta, "Numerical linear algebra and applications," Brokes/Cole publishing company, 1994.
- [6] S. Duan, "Stability of uncertain piecewise affine time-delay systems with application to all wheel drive clutch control," *Mechanical Engineering*, pp.82, 2011
- [7] H. Gorecki, S. Fuksa, P. Grabowski and A. Korytowski, "Analysis and Synthesis of Time Delay Systems," John wiley & Sons, New York, 1989
- [8] K. Gu, S. I. Niculescu, "Stability analysis of time-delay systems: A Lyapunov approach," *Advanced Topics in Control Systems Theory, Lecture Notes in Control and Information Sciences* 328, pp.139-170, 2006.
- [9] J. K. Hale and S. M. V. Lunel, "Introduction to Functional Differential Equations," Springer, New York, 1993.
- [10] I. Ivanoviene and J. Rimas, "Analysis of delay differential equations using the lambert function," *Mathematical Modelling and Analysis*, pp.1-9, 2010.
- [11] E. Jarlebring and T. Damm, "The Lambert W function and the spectrum of some multidimensional time-delay systems," *Automatica*, pp.1-5, 2007.
- [12] P. L. Liu, "Exponential stability for linear time-delay systems with delay dependence," *Journal of the Franklin Institute* 340(6-7), pp.481-488, 2003.
- [13] F. Maghami Asl and A. Galip Ulsoy, "Analysis of a system of linear delay differential equations," *Journal of Dynamic Systems*, pp.1-9, 2003.
- [14] M. Malek-Zavarei and M. Jamshidi, "Time-delay systems: analysis, optimization, and applications," Elsevier Science Pub., New York, U.S.A, 1987.
- [15] A. Manitius and A. W. Olbrot, "Finite spectrum assignment problem for systems with delays," *IEEE Transactions on Automatica Control* 24(4), pp.541-553, 1993.
- [16] W. Michiels, K. Engelborghs, P. Vansevenant, and D. Roose, "Continuous pole placement for delay equations," *Automatica* 38(5), pp.747-761, 2002.
- [17] A. W. Olbrot, "On controllability of linear systems with time delays in control," *IEEE*

Transactions on Automatica Control 17(5), pp.664-666, 1972.

- [18] M. C. Pease, “*Methods of Matrix Algebra*,” Academic Press, New York, 1965.
- [19] H. Radjavi and P. Rosental, “*Simultaneous Triangularization*,” Springer, New York, 2000.
- [20] J. P. Richard, “Time-delay systems : an overview of some recent advances and open problems,” *Automatica* 39(10), pp.1667-1964, 2003.
- [21] H. Shinozaki, “Lambert W function approach to stability and stabilization problems for linear time-delay systems,” *Kyoto Institute of Technology*, pp.5-21, 2007.
- [22] H. Shinozaki and T. Mori, “Robust stability analysis of linear time-delay systems by lambert W function,” *Automatica* 42(10), pp.1791-1799, 2006.
- [23] G. J. Silva, A. Datta and S. P. Bhattacharyya, “Controller design via pade approximation can lead to instability,” in *Proceedings of the 40th IEEE Conference on Decision and Control*, pp.4733-4737, 2001.
- [24] O. J. Smith, “Closer control of loops with dead time,” *Chem. Eng. Prog.*, 53, pp.217-219, 1957.
- [25] W. Wiboonjaroen and S. Sujitjron, “Stabilization of State-derivative Feedback Control with Time Delay,” *Research Journal of Applied Sciences, Engineering and Technology* 4(18), pp.3201-3208, 2012.
- [26] E. M. Wright, “Solution of the equation $ze^z = a$,” *Bull. Am. Math. Soc*, pp.65, 1965.
- [27] S. Yi, P. W. Nelson and A. G. Ulsoy, “Eigenvalue assignment via the lambert W function for control of time-delay systems,” *Journal of Vibration and Control*, pp.1-22, 2010.
- [28] S. Yi, P. W. Nelson and A. G. Ulsoy, “Chatter Stability analysis using the lambert function and bifurcation analysis,” *Mathematical Biosciences and Engineering* 4(2), pp.355-38, 2007b.
- [29] S. Yi, P. W. Nelson and A. G. Ulsoy, “Controllability and observability of systems of linear delay differential equations via the matrix lambert function,” *IEEE Transactions on Automatica Control*, pp.1-10, 2008.
- [30] S. Yi, P. W. Nelson and A. G. Ulsoy, “Robust Control and Time-Domain Specifications for Systems of Delay Differential Equations via Eigenvalue Assignment,” *Journal of Dynamic Systems, Measurement and Control*, vol.132, pp.1-10, 2010.
- [31] S. Yi, P. W. Nelson and A. G. Ulsoy, “Survey on analysis of time delayed systems via the lambert W function,” *Dynamics of Cotinuous*, pp.1-6, 2007.
- [32] S. Yi, “Time-delay systems: analysis and control using the lambert W function,” *Mechanical Engineering*, pp.1-193, 2009.
- [33] Q. C. Zhong, “*Robust control of time-delay systems*,” Springer, London, 2006.

[۳۴] ع. خاکی صدیق، "سیستم های کنترل خطی"، چاپ دوم، انتشارات دانشگاه پیام نور، ص. ۸۱،

۱۳۸۱.