

ỨNG DỤNG CỦA HÀM LAMBERT W ĐỐI VỚI MỘT DẠNG PHƯƠNG TRÌNH, HỆ PHƯƠNG TRÌNH VI PHÂN CÓ TRỄ

Hoàng Việt Anh*, Nguyễn Thị Hương Lan,
Trường Đại học Tây Bắc

Tóm tắt: Bài viết này tổng quan về phương pháp tiếp cận hàm Lambert W, để phân tích và điều khiển các hệ điều khiển tuyến tính, hệ số hằng với đơn trễ. Giải pháp thông qua hàm Lambert W được trình bày cho các hệ bậc một, sau đó được mở rộng cho các hệ bậc cao hơn bằng cách sử dụng ma trận hàm Lambert W. Việc sử dụng phần mềm mã nguồn mở dựa trên gói công cụ LambertW_DDE trong Matlab được giới thiệu bằng các ví dụ.

Từ khóa: Thời gian trễ, phương trình vi phân trễ, hàm Lambert W, phần mềm.

1. GIỚI THIỆU

Hệ điều khiển có trễ nảy sinh trong rất nhiều bài toán trong tự nhiên và kỹ thuật, chẳng hạn như trong các quá trình với trễ vận chuyển, các bài toán dòng giao thông, các hệ sinh học, viễn thông và còn rất nhiều vấn đề khác. Bài báo tập trung vào một cách tiếp cận đặc biệt và được phát triển gần đây, dựa trên cách tiếp cận và sử dụng hàm Lambert W, để phân tích và điều khiển các hệ điều khiển tuyến tính, hệ số hằng với đơn trễ

2. ỨNG DỤNG CỦA HÀM LAMBERT W TRONG LÝ THUYẾT PHƯƠNG TRÌNH VI PHÂN CÓ TRỄ

2.1. Trường hợp vô hướng

Trước hết chúng ta xét hệ điều khiển trong đó cả biến điều khiển $u(t)$ và biến trạng thái $x(t)$ đều là vô hướng

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

trong đó a, b, a_d là các hằng số đã biết và h là trễ hằng số. Điều kiện ban đầu $x(0) = x_0$ và $x(t) = g(t)$ với mọi $-h \leq t < 0$ phải được cho

trước. Hàm Lambert W được áp dụng để giải phương trình đặc trưng được viết lại dưới dạng

$$(s-a)e^{sh} = a_d. \quad (2)$$

Nhân cả hai vế của phương trình (2) với he^{-ah} ta nhận được

$$h(s-a)e^{h(s-a)} = a_d h e^{-ah}. \quad (3)$$

Dựa trên định nghĩa của hàm Lambert W trong biểu thức (3) ta có

$$W(a_d h e^{-ah}) e^{W(a_d h e^{-ah})} = a_d h e^{-ah}. \quad (4)$$

So sánh các phương trình (3) và (4) ta có

$$h(s-a) = W(a_d h e^{-ah}). \quad (5)$$

Do đó, nghiệm của phương trình đặc trưng trong phương trình (2) có thể được viết theo hàm Lambert W như sau:

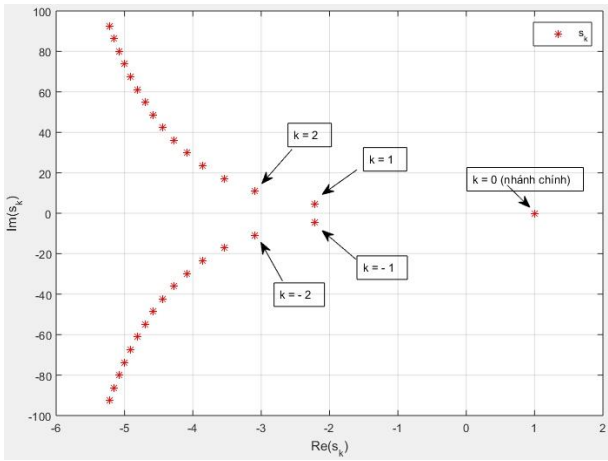
$$s = \frac{1}{h} W(a_d h e^{-ah}) + a. \quad (6)$$

Phổ của phương trình DDE vô hướng (1) thu được bằng cách sử dụng vô hạn nhánh của hàm Lambert W và được trình bày tường minh theo các tham số a, a_d và h của hệ. Các nghiệm của phương trình đặc trưng (2), cho $k = 0, \pm 1, \pm 2, \dots, \pm \infty$ là

$$s_k = \frac{1}{h} W_k(a_d h e^{-ah}) + a. \quad (7)$$

Hơn nữa, đối với phương trình (1) tích chất ổn định được xác định bởi giá trị riêng bên phải (nửa phỏ phải) của mặt phẳng phức (tức là, $k=0$). Do đó, để đảm bảo sự ổn định, không cần thiết phải kiểm tra các giá trị riêng khác trong phỏ vô hạn.

Ví dụ 1: Cho $a = -1, a_d = 0.5$, và $h = 1$ các nghiệm đặc trưng thu được bằng phương trình (7) và hàm Lambert W trong MATLAB, và được vẽ trong Hình 1.



Hình 1: Giá trị riêng của phương trình (1) khi $a = -1, a_d = 0.5$, và $h = 1$. Giá trị riêng bên phải là tương ứng với nhánh chính $k = 0$, các cặp tiếp theo lần lượt tương ứng với $k = \pm 1, \pm 2, \dots, \pm 15$.

Trong [6] nghiệm tổng quát của phương trình (1) được biểu diễn dưới dạng một chuỗi vô hạn dựa trên các giá trị riêng trong công thức (7) như sau

$$x(t) = \sum_{k=-\infty}^{+\infty} e^{s_k t} C_k^I + \int_0^t \sum_{k=-\infty}^{+\infty} e^{s_k(t-\eta)} C_k^N b u(\eta) d\eta, \quad (8)$$

trong đó

$$C_k^I = \frac{x_0 + a_d e^{-s_k h} \int_0^h e^{-s_k t} g(t-h) dt}{1 + a_d h e^{-s_k h}}, \quad (9)$$

và

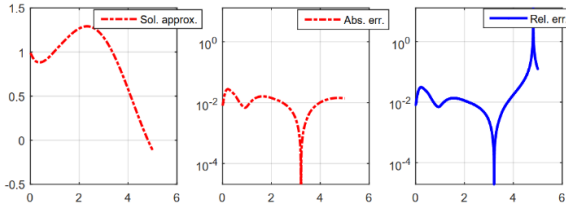
$$C_k^N = \frac{1}{1 + a_d h e^{-s_k h}}. \quad (10)$$

Lưu ý rằng các hệ số C_k^I được xác định từ hàm $g(t)$ và trạng thái ban đầu x_0 , và các hệ số C_k^N chỉ được xác định theo các tham số hệ a, a_d và h . Do đó, nghiệm tổng quát của phương trình (8) có thể được xem là tổng của một nghiệm của hệ thuần nhất (mà chúng ta sẽ gọi là *nghiệm tự do (free solution)*) và một nghiệm riêng của hệ không thuần nhất (mà chúng ta sẽ gọi là *nghiệm chịu lực tác động (forced solution)*).

Các điều kiện để hội tụ của các nghiệm dạng chuỗi như vậy được thảo luận bởi Bellman và Cooke. Ví dụ, nếu tất cả mọi giá trị riêng s_k đều có phần thực âm và tồn tại một chặn dưới dương cho khoảng cách giữa các giá trị riêng thì chuỗi vô hạn nói trên sẽ hội tụ. Một khía cạnh khác rất quan trọng của biểu diễn dạng chuỗi này của nghiệm $x(t)$ đó là việc chặt cụt chuỗi, ví dụ $k = 0, \pm 1, \pm 2, \dots, \pm n$, sẽ dẫn đến một xấp xỉ cho nghiệm $x(t)$ dựa trên tối đa $(2n+1)$ giá trị riêng ngoài cùng bên phải (còn được gọi là các giá trị riêng trội). Hệ quả, ta có thể xây dựng một biểu diễn xấp xỉ hữu hạn chiều cho hệ (1) mà vẫn giữ nguyên được các tính chất động lực của hệ.

Ví dụ 2: (Xấp xỉ phản hồi trong trường hợp vô hướng) Cho $a = -1, a_d = 0.5$, $b = 1$ và $h = 1$ ta có thể tìm được các giá trị của s_k thông qua biểu thức (7) và hàm Lambert W như trong Ví

dự 1 và các giá trị của C_k^I và C_k^N sử dụng các phương trình (9) - (10), trong đó $x_0 = 1$ và $g(t) = 1$ với $-h \leq t < 0$. Các giá trị riêng s_k và các hệ số này được trình bày trong Bảng 1. Hình 2 thể hiện phản hồi cho trường hợp $u(t) = \sin(t)$ và so sánh giữa phương pháp dựa trên hàm Lambert W (sử dụng 7 thành phần trong Bảng 1) và nghiệm số (sử dụng hàm dde23 trong MATLAB).



Hình 2: Phản hồi cho trường hợp

$u(t) = \sin(t)$, với $x_0 = 1$ và $g(t) = 1$ cho

$-h \leq t < 0$, và so sánh giữa công thức chặt

cụt nghiệm giữ 7 thành phần đầu tiên dựa

trên phương pháp hàm Lambert (xem Bảng 1)

và phương pháp số (hàm dde23 trong

MATLAB). Các hệ số là $a = -1, a_d = 0.5$, và

$$h = 1.$$

k	s_k	C_k^I	C_k^N
0	-0.3149	1.0713	0.5934
± 1	$-2.2211 \pm 4.4442 i$	$-1.9829 \pm 0.4711 i$	$-0.0112 \pm 0.2245 i$
± 2	$-3.0915 \pm 10.8044 i$	$-2.0109 \pm 0.1863 i$	$-0.0093 \pm 0.0916 i$
± 3	$-3.5450 \pm 17.1313 i$	$-2.0073 \pm 0.1168 i$	$-0.0052 \pm 0.0579 i$
± 4	$-3.8550 \pm 23.4407 i$	$-2.0050 \pm 0.0852 i$	$-0.0034 \pm 0.0424 i$
± 5	$-4.0911 \pm 29.7416 i$	$-2.0036 \pm 0.0671 i$	$-0.0024 \pm 0.0335 i$

Bảng 1: Các giá trị riêng và hệ số trong Ví dụ 2.

2.2. Trường hợp hệ phương trình

Phép xấp xỉ được trình bày trong phần trước đã được khái quát hóa trong [7] cho hệ điều khiển có trễ có dạng sau

$$\begin{aligned} \dot{x}(t) &= Ax(t) + A_d x(t-h) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \quad (11)$$

trong đó $x(t)$ là vectơ trạng thái, $u(t)$ là biến điều khiển, $y(t)$ là vectơ đầu ra, A, A_d, B, C và D là các ma trận hệ số và h là độ trễ vô hướng đã biết. Điều kiện ban đầu $x(0) = x_0$ và $x(t) = g(t)$ với $-h \leq t < 0$ phải được cho trước. Công thức nghiệm của hệ (1) có dạng:

$$x(t) = \sum_{k=-\infty}^{\infty} e^{S_k t} C_k^I + \int_0^t \sum_{k=-\infty}^{\infty} e^{S_k(t-\eta)} C_k^N B u(\eta) d\eta, \quad (12)$$

trong đó

$$S_k = \frac{1}{h} W_k(A_d h Q_k) + A, \quad (13)$$

và Q_k được lấy từ nghiệm số (bằng cách sử dụng hàm `fsolve` trong MATLAB) của phương trình

$$W_k(A_d h Q_k) e^{W_k(A_d h Q_k) + Ah} = A_d h. \quad (14)$$

Các đại lượng Q_k, W_k, S_k, C_k^I và C_k^N trong các phương trình (12) - (14) đều có thể tính được bằng các lệnh trong gói công cụ LambertWDDE theo các điều kiện đã cho $h, A, A_d, g(t), x_0, B$ và $u(t)$. Các hàm chính của gói công cụ LambertWDDE được tóm tắt trong Bảng 2.

Tên	Mô tả
<code>lambertw_matrix</code>	Tính hàm ma trận LambertW
<code>find_Sk</code>	Tìm S_k và Q_k cho một nhánh cụ thể
<code>find_CI</code>	Tính C_I cho một nhánh cụ thể trong điều kiện ban đầu
<code>find_CN</code>	Tính C_N cho một nhánh cụ thể
<code>pwcont_test</code>	Thử tính điều khiển được cho hệ có trễ
<code>pwobs_test</code>	Thử tính quan sát được cho hệ có trễ
<code>cont_gramian_dde</code>	Ma trận điều khiển Gramian của hệ có trễ
<code>obser_gramian_dde</code>	Ma trận quan sát Gramian của hệ có trễ
<code>place_dde</code>	Phân bố các giá trị riêng trội của hệ có trễ
<code>stabilityradius_dde</code>	Tính bán kính ổn định cho hệ có trễ

Bảng 2: Các hàm chính của gói công cụ LambertWDDE

Để có được S_k trong một trường hợp k cụ thể, ta cần giải phương trình (14) để tìm Q_k trước, tiếp theo ta thay thế kết quả vào biểu

thức (13) để lấy được S_k . Các bước này được thực hiện trong hàm $find_Sk$. Lưu ý rằng, hàm ma trận Lambert W_k trong biểu thức (13) và (14) được tính toán bằng hàm $lambertw_matrix$, dựa trên phép biến đổi dạng Jordan. Để hiểu rõ hơn toàn bộ quá trình, chúng ta tìm hiểu trong ví dụ sau.

Ví dụ 3: Cho

$$A = \begin{bmatrix} -1 & -3 \\ 2 & -5 \end{bmatrix}, A_d = \begin{bmatrix} 1.66 & -0.697 \\ 0.93 & -0.33 \end{bmatrix}, h = 1.$$

Để tìm các ma trận S_k chúng ta sẽ sử dụng hàm $find_Sk$ trong gói công cụ LambertWDDE.

Đối với nhánh chính, $k = 0$, ta có được:

$$S_0 = \begin{bmatrix} 0.3055 & -1.4150 \\ 2.1317 & -3.3015 \end{bmatrix}.$$

Các giá trị riêng của S_0 là -1.0119 và -1.9841 . Tiếp theo, sử dụng $find_CI$ và $find_CN$, ta có thể nhận được các hệ số cho nghiệm dạng chuỗi của phương trình (12). Ví dụ, nếu ta chọn $u(t) = 0, g(t) = 0$, và ta có một bước nhảy tại $t = 0$ đến $x_0 = [1 \ 1]^T$, ta có thể nhận được (sử dụng $find_CI$) hệ số cho phản hồi tự do cho $k = 0$ là:

$$C_0^I = \begin{bmatrix} 0.2635 \\ 0.4290 \end{bmatrix}.$$

Do đó, phép xấp xỉ sử dụng một nhánh $k = 0$ cho phản hồi tự do là:

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \exp\left(\begin{bmatrix} 0.3055 & -1.4150 \\ 2.1317 & -3.3015 \end{bmatrix} t\right) \begin{bmatrix} 0.2635 \\ 0.4290 \end{bmatrix}.$$

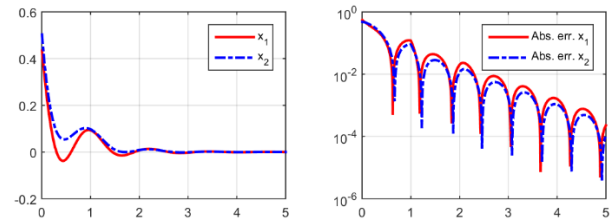
Ở đây ta sử dụng ký hiệu exp để chỉ hàm mũ cho cả trường hợp vô hướng lẫn ma

trận. Lưu ý rằng trong MATLAB, hàm mũ của ma trận là hàm $expm$, chứ không phải hàm vô hướng exp . Để cải thiện phép xấp xỉ, quy trình này có thể được lặp lại cho các nhánh bổ sung thứ k , sau đó sự xấp xỉ dạng chuỗi của nghiệm có thể thu được thông qua phương trình (12) (xem Hình 3). Ví dụ, các nhánh $k = \pm 1$ cho ta các ma trận S_k liên hợp phức bổ sung:

$$S_{-1,+1} = \begin{bmatrix} -0.3499 \pm 4.980i & -1.6253 \pm 0.1459i \\ 2.4174 \pm 0.1308i & -5.1048 \pm 4.5592i \end{bmatrix},$$

với các hệ số liên hợp phức cho phản hồi tự do với $k = \pm 1$ là:

$$C_{-1,+1}^I = \begin{bmatrix} 0.0909 \pm 0.1457i \\ 0.0435 \pm 0.1938i \end{bmatrix}.$$



Hình 3: Phản hồi gần đúng (3 thành phần) và sai số tuyệt đối cho hệ trong Ví dụ 3 so với nghiệm sử dụng hàm dựng sẵn dde23 trong MATLAB.

KẾT LUẬN

Ứng dụng của hàm LambertW trong lý thuyết phương trình vi phân có trễ. Đặc trưng của phương pháp này là sử dụng các nhánh của hàm LambertW để biểu diễn nghiệm dưới dạng một chuỗi vô hạn với biến số là các giá trị riêng của hệ điều khiển có trễ đang xét.

Bằng việc sử dụng hàm LambertW, chúng ta đưa ra được công thức nghiệm tường minh của hệ có trễ trong cả hai trường hợp nghiệm tự do và nghiệm chịu lực tác động.

Phần mềm nguồn mở trong gói công cụ Lambert WDDE, cũng như các ví dụ kèm theo, được hy vọng sẽ làm cho cách tiếp cận dựa trên hàm Lambert W trở nên dễ tiếp cận và hữu ích hơn cho những người quan tâm đến các ứng dụng được mô hình hóa dưới dạng hệ điều khiển có trễ hằng số.

TÀI LIỆU THAM KHẢO

- [1] C. T. Chen. *Linear System Theory and Design*. Oxford University Press, 1998.
- [2] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth. On the Lambert W function. In: *Advances in Computational Mathematics*, 5:329–359, 1996.
- [3] S. Duan. Supplement for time delay systems: analysis and control using the Lambert W function. <http://www.umich.edu/ulsoy/tds-supplement.htm>. Technical report, 2010.
- [4] M. Malek-Zavarei and M. Jamshidi. *Time-Delay Systems: Analysis, Optimization and Applications*. New York, USA: Elsevier Science Pub., 1987.
- [5] S. Yi, S. Duan, P. Nelson, and A. Ulsoy. The Lambert W function approach to time delay systems and the LambertW-DDE toolbox. *IFAC Proceedings Volumes*, 45(14):114 – 119, 2012. 10th IFAC Workshop on Time Delay Systems.
- [6] S. Yi, P. W. Nelson, and A. G. Ulsoy. Eigenvalue assignment via the Lambert W function for control of time-delay systems. *J. Vibration and Control*, 16(7-8):961-982, 2010.
- [7] S. Yi, P. W. Nelson, and A. G. Ulsoy. *Time-Delay Systems: Analysis and Control Using the Lambert W Function*. World Scientific, 2010.

A STUDY OF APPLYING THE LAMBERT W TO DELAYED DIFFERENTIAL EQUATION THEORY

Hoang Viet Anh*, Nguyen Thi Huong Lan

Tay Bac University

Abstract: Currently, the theory of differential equations has many applications in practice and in life. This article reviews the applications of Lambert W function in delayed differential equation theory. The Lambert W scalar function has been pre-programmed in many scientific computing software, such as the Lambertw function in MATLAB. These functions are useful in combinatorics (for example, tree enumeration) as well as in relativity and quantum mechanics. They can be used to solve a wide variety of equations containing up to exponents (e.g. maxima of the Planck, Bose-Einstein and Fermi-Dirac distributions) as well as in solving delayed differential equation theory.

Keywords: delayed, delayed differential equation theory, Lambert W function.

Ngày nhận bài: 14/03/2022. Ngày nhận đăng: 25/05/2022.

Liên lạc: Hoàng Việt Anh, e - mail: hoangvietanh@utb.edu.vn