AN OBSERVER FOR STATE ESTIMATION IN THE PRESENCE OF MULTIPLE MEASUREMENT DELAYS

A THESIS

Presented to the Department of Mechanical and Aerospace Engineering

California State University, Long Beach

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Aerospace Engineering

Committee Members:

Praveen Shankar, Ph.D. (Chair)
Panadda Marayong, Ph.D.
Emel Demircan, Ph.D.

College Designee:

Hamid Rahai, Ph.D.

By Jeffrey K. Won

B.S., 2015, University of California, Davis

May 2019

ABSTRACT

AN OBSERVER FOR STATE ESTIMATION IN THE PRESENCE OF MULTIPLE MEASUREMENT DELAYS

By

Jeffrey K. Won

May 2019

A major challenge of implementing wireless sensor networks in the feedback control of dynamic systems is dealing with the time delays inherent in the wireless network. If not properly addressed, these delays can cause instabilities and failures in the control system. This thesis proposes a remedy to this problem: a Delay-Compensating Estimator (DCE) which uses the delayed measurements, along with the known linearized dynamics of the system being measured, to predict the current, un-delayed state of the system. This is achieved via a Luenberger observer with an induced delay in the observer feedback which matches the expected delay of the sensors, similar in concept to the Smith predictor. The design methodology for the DCE is expounded on, and delay robustness is considered in comparison to a classic Luenberger observer. Finally, an application of the DCE to real-time control of a double inverted pendulum on a cart is presented.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Dr. Praveen Shankar, for his invaluable guidance and support, and without whom this thesis would not have been possible. I would also like to thank my committee members, Dr. Panadda Marayong and Dr. Emel Demircan, for reviewing my thesis. Lastly, I am eternally grateful to my family for their unwavering love and support for me.

TABLE OF CONTENTS

ABSTRACT ii

ACKNOWLEDGMENTS iii

LIST OF TABLES v

LIST OF FIGURES vi

1. INTRODUCTION 1

2. BACKGROUND AND LITERATURE REVIEW 4

3. ESTIMATOR DESIGN USING THE LAMBERT W METHOD 12

4. EFFECTS OF DELAY UNCERTAINTY ON ESTIMATOR DESIGN 24

5. APPLICATION TO THE CONTROL OF A DYNAMIC SYSTEM USING ESTIMATED STATE FEEDBACK WITH MEASUREMENT DELAY 38

6. CONCLUSION AND FUTURE WORK 50

APPENDICES 52

A. INFINITE ROOT SPECTRUM OF A TIME DELAY SYSTEM 53

B. EQUIVALENCE OF MEASUREMENT DELAYS AND CONTROLLER INPUT DELAYS IN IDEAL FEEDBACK CONTROL SYSTEMS 55

C. SOLUTION OF INTEGRAL (4.12) 57

D. DERIVATION OF THE DYNAMICS OF A DOUBLE INVERTED PENDULUM ON A CART 59

REFERENCES 63

LIST OF TABLES

1 Summary of Equations for the Lambert W Method 15

LIST OF FIGURES

2.1 Branch cuts of the Lambert W function.. 8

3.1 Block diagram of the proposed Delay-Compensating Estimator (DCE)........................ 13

3.2 Simulated DCE performance for a mass-spring-damper system 22

4.1 Block diagram of the estimator with uncertainty in delay modeling 25

4.2 Critical values of τ_r below which the estimator is considered failed............................ 30

4.3 DCE performance with delay modeling errors 34

4.4 MERF for the damped oscillator.. 35

4.5 MERF for the undamped oscillator 36

5.1 Schematic of the double inverted pendulum on a cart.. 40

5.2 Main simulation block diagram in Simulink.. 44

5.3 Internal block diagram of the estimator.. 45

5.4 System response of Case 1 (ideal DCE) 46

5.5 System response of Case 2 (small delay uncertainty) 47

5.6 System response for Case 3 (large delay modeling errors)... 47

5.7 System response for Case 4 (no real delay) 48

5.8 System response with a Luenberger observer instead of the DCE................................ 49

CHAPTER 1

INTRODUCTION

Time delays are unavoidable in a wide range of systems, including mechanical, electrical, and chemical processes [1]. It is thus necessary to account for them when modeling the system dynamics and designing a control system. However, time delays pose an especially difficult mathematical challenge because they cause the impacted systems to have an infinite number of roots in the Laplace domain [2] (see Appendix A), and thus confound most conventional root-finding and root-placement techniques.

One common cause of time delays in feedback control systems is in the measurement process, especially with the advent of digitally processed sensor signals and wireless sensor networks, which are particularly susceptible to delays [3]. The delays inherent in these measurements can significantly affect the control system performance if not accounted for, so an increasing amount of literature has been focused on the design of controllers for systems with time delays (e.g., [4-7]).

When designing a controller for a system affected by such delays, it is often expedient to lump the measurement delay together with any controller actuation delay, as they are indistinguishable in terms of the ideal closed-loop dynamics of the system (see Appendix B for proof). However, in some cases, it can be advantageous to model the measurement delays separately from the controller delays.

Take, for example, target tracking applications. This is purely an estimation problem, with no controller to be implemented. Time delays can be a major source of tracking error in cases where sonar is used as the tracking measurement [8], but the target tracking accuracy can be significantly improved when the time delays are properly accounted for.

Other cases when only the measurement is needed include scientific studies and real-time diagnostics. In these cases, too, there is no controller to be designed around the measurement delay, but one would still like to have the un-delayed measurements in real time.

Another case is when measurements are taken at multiple locations in a system, each with a different delay. These differing measurement delays can present an incoherent picture of the state of the system at any given time. As an example, consider the method of real-time bending mode reconstruction using distributed strain sensors studied in [9]. This method relies on strain measurements taken at each timestep to build a picture of the bending modes of the structure. Rezaei et al. [10] showed that the method suffers greatly when each strain sensor has a different measurement delay, because then the strain measurements observed at any given timestep are not coherent. Even if the end goal were to design a controller to damp these vibration modes, the problem is much simpler to tackle if taken in parts: first determine the true mode shapes by compensating for the different measurement delays, and then design a controller around the control input delay under the assumption that the mode shapes are known in real time.

Problem Formulation

The primary focus of this thesis is the development of a Delay-Compensating Estimator (DCE) which can compensate for measurement delays to produce a predicted un-delayed measurement in real time. We will consider the following linear time-invariant (LTI) system with multiple measurement delays:

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

$$y(t) = \sum_{i=1}^{p} C_i x(t - \tau_i) \tag{1.1}$$

where $x(t)$ is the $n \times 1$ state vector, $u(t)$ is the control input A and B are the system and input matrices, $y(t)$ is the $m \times 1$ measurement vector, and C_i, τ_i $(i = 1,2, \ldots p)$ are the output matrices for each measurement and their associated delays, respectively. The delays τ_i are assumed to be constant, but may or may not be known precisely.

The major original contribution of this work is the theoretical development of the estimator architecture and design process, as well as limits on its applicability in delay space. A secondary contribution is the extension of a current method for analysis of single-time-delay systems to handle multiple delays.

This thesis is organized as follows: Chapter 2 presents background information and a review of literature on time delay systems and observers/estimators, Chapter 3 formulates the estimator and presents the design method, Chapter 4 assesses the estimator performance in the presence of uncertainties in the delays, Chapter 5 presents an example application of the estimator to the control of a double inverted pendulum on a cart, and finally, Chapter 6 provides concluding remarks.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

Observers and Estimators

Although the terms "observer" and "estimator" are often used interchangeably, they tend to have slightly different connotations. "Observer" is more often used for deterministic state reconstruction, while "estimator" is typically the term used for measurement of stochastic systems. In light of this, although the delay-compensating estimator (DCE) is formulated for a deterministic system, its function is more akin to that of an estimator in that it attempts to filter out measurement errors due specifically to time delays in the measurements. Additionally, as will be shown in Chapter 3, the DCE cannot actually reconstruct a state vector from few measurements the way that a classic Luenberger observer [11] might. While the Luenberger observer is able to reconstruct unobserved states from limited measurements, the DCE requires all states to be measurable in some form in order to work. However, as implied by the name, the DCE is able to compensate for measurement delays to provide the un-delayed state of the system, which is something the Luenberger observer cannot do.

Despite this difference in function, the similarity in form means it is still useful to look at methods of designing the estimator gain L for the classic Luenberger observer,

$$\begin{gathered}\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L[y(t) - \hat{y}(t)] \\ \hat{y}(t) = C\hat{x}(t). \end{gathered} \tag{2.1}$$

Design methods for state-feedback controllers such as robust pole placement [12] or linear quadratic regulator (LQR) are easily adapted for this observer by simply transposing the inputs and outputs to the design algorithm, i.e. $A = A^T,\ B = C^T, F = L^T$. Doing so will impart the designated response (either the desired poles or the minimum weighted cost) to the observer error function

$$e_x(t) = x(t) - \hat{x}(t).$$

A similar concept can be applied to the design of the DCE gain matrix. The idea is that, if we can find an established method for controller design for time-delay systems, it should be easily adaptable to fit the DCE.

Time-Delay Systems

One of the earliest controllers for time-delay systems is the Smith predictor, first proposed by O. J. M. Smith in 1957 [13]. It consists of a model of the controlled system being run in parallel to the real system, with the delayed and un-delayed model outputs being used in the feedback loop to essentially cancel out the effect of the delay on the total transfer function. The Smith predictor has received significant research attention [14], but remains highly sensitive to uncertainties in the delay [15].

More recent advances in time-delay control include robust adaptive controllers such as the one proposed by Hua et al. [5], which stabilizes an uncertain system with process delay (instead of feedback delay). While this method guarantees stability, it does not guarantee performance, which is often a more important requirement for estimators than simple stability.

A partial pole placement method based on numerical optimization of the un-placed poles was developed by Michiels et al. [6]. The method is similar in principle to the robust pole placement algorithm of Kautsky et al. [12], but handles time delays appropriately. This offers a powerful design option for single-input, single-output (SISO) systems, but does not handle the more general case of multiple-input systems.

A robust observer for time-delay systems with unknown inputs was formulated by Sename [16]. This observer was designed to decouple the effects of the unknown input from the estimation error. However, while the observer is guaranteed to be stable (that is, it will eventually converge to the correct state even in the presence of unknown inputs), no claims are made about its rate of convergence. This can pose a problem if there are specific requirements for the transient performance of the estimator, as Sename's design method does not take performance into account at all.

A notable recent advance in time-delay analysis by Olgac and Sipahi [17] enables the generation of a complete stability map of a system with respect to changes in the time delays. This method makes use of an algebraic substitution which transforms the infinite-dimensional root spectrum of a time-delay system into a finite spectrum which periodically generates the rest of the roots. The substitution is valid only on the imaginary axis, so the method can only be used to analyze the points at which the system crosses from stability to instability. Therefore, while it is a powerful method for designing stable controllers which are robust to delay uncertainties, it offers no way of analyzing the performance or convergence rate of the controller. The method is also applicable to systems with multiple delays via the extension discussed in [18].

Another notable work was the development of an analytic solution for time-delay equations using the Lambert W function by Asl and Ulsoy [19], which was then extended to the general matrix case by Yi and Ulsoy [20]. Unlike previous numerical solutions, this solution produces the full eigenvalue spectrum of the time delay system, which makes controller design via eigenvalue assignment a real possibility [4]. This method was only formulated for systems with a single time delay.

While several methods for time-delay controller design have been developed, each has its own limitations and drawbacks. The most promising method for the specific application of designing the estimator gain for the DCE is the Lambert W method developed by Yi & Ulsoy [20], as it can handle multiple-input multiple-output (MIMO) systems of arbitrary size, and it allows time-domain performance specification via rightmost eigenvalue assignment. The main limitations of the method are that it requires a LTI system, and that it does not handle multiple delays. The DCE is formulated for a single delay, so the LTI limitation is of no concern, and the method can be readily extended to multiple delays, as will be shown in Chapter 3.

Since the Lambert W method will be the method of choice for our estimator gain design process, we would do well to briefly discuss its namesake, the Lambert W function.

The Lambert W Function

The Lambert W function, as described by Corless et al. [21], is defined as the function $w(z)$ which satisfies

$$w(z)e^{w(z)} = z.$$

In general, $w(z)$ is multi-valued in complex space, with an infinite number of branches $w_k(z)$, where $k = -\infty, \ldots -1, 0, 1, \ldots \infty$. The branch cuts are shown in Figure 2.1, taken directly from [21]. The scalar Lambert W function can be calculated via a power series,

$$w_0(z) = \sum_{n=1}^{\infty} \frac{(-n)^{n-1}}{n!} x^n$$

or via Newton iteration, and is implemented in MATLAB as `lambertw(k,z)` [22]. Similarly, the matrix Lambert W function $W_k(Z)$ is defined as the matrix function which satisfies

$$W_k(Z)e^{W_k(Z)} = Z. \tag{2.2}$$

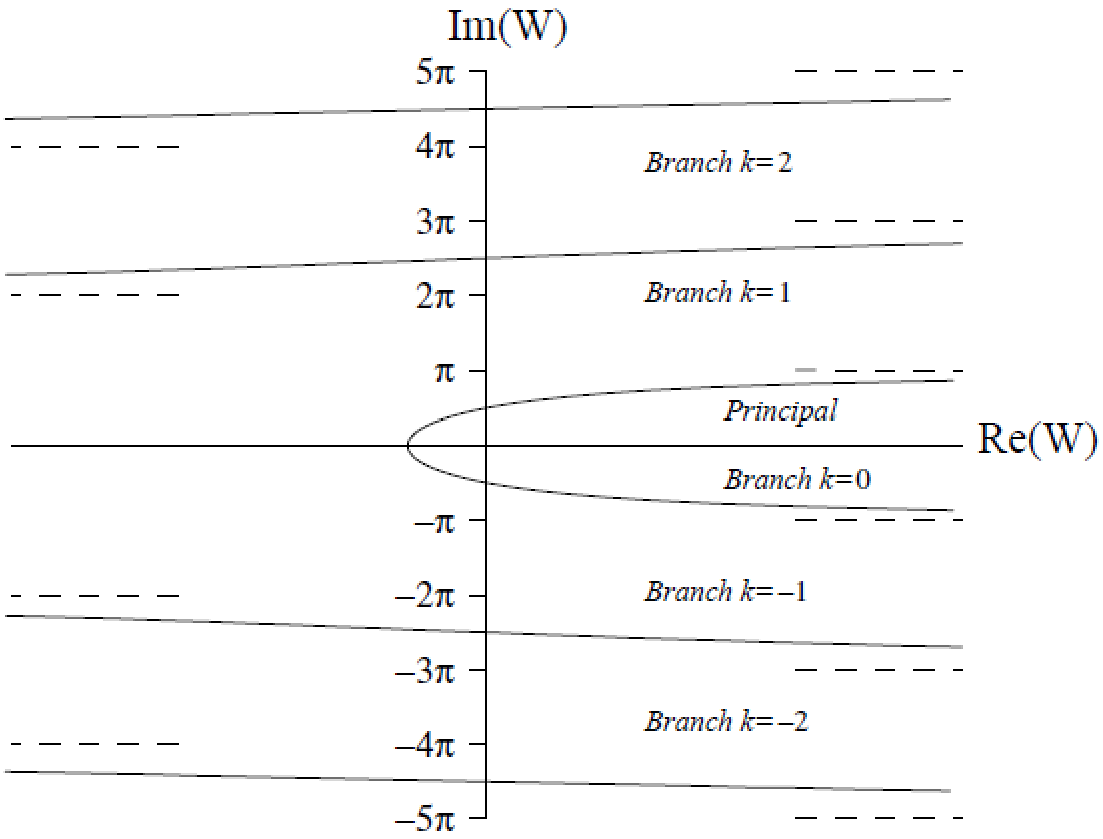

FIGURE 2.1. Branch cuts of the Lambert W function [21].

An efficient numerical procedure for calculating $W_k(z)$ based on Schur decomposition and Newton iteration was proposed by Fasi, et al. [23] and is the algorithm used in this work.

The Lambert W Method for Single Delays

Yi [24] derives the solution to an n-dimensional LTI system of delay differential equations (DDEs) of the form

$$\dot{x}(t) = Ax(t) + A_d x(t - \tau), \tag{2.3}$$

where A_d is the $n \times n$ system matrix for the delayed state, as follows. First, assume a solution of the form

$$x(t) = e^{St}\Gamma \tag{2.4}$$

where S is an $n \times n$ matrix and Γ is an $n \times 1$ constant vector based on the initial conditions. Substituting (2.4) into (2.3) yields

$$Se^{St}\Gamma - Ae^{St}\Gamma - A_d e^{S(t-\tau)}\Gamma = 0.$$

Collecting terms gives

$$(S - A - A_d e^{-S\tau})e^{St}\Gamma = 0. \tag{2.5}$$

Because (2.5) must be satisfied for any initial condition and for all time t, it can be concluded that

$$S - A - A_d e^{-S\tau} = 0.$$

Post-multiplying by $\tau e^{S\tau}e^{-A\tau}$ and rearranging, we get

$$\tau(S - A)e^{S\tau}e^{-A\tau} = A_d \tau e^{-A\tau}. \tag{2.6}$$

Since S and A do not commute in general, we cannot say that $e^{S\tau}e^{-A\tau} \neq e^{(S-A)\tau}$ and the Lambert W function cannot be directly substituted. Therefore, an unknown matrix Q is introduced which satisfies the equation

$$\tau(S - A)e^{(S-A)\tau} = A_d \tau Q.$$

Using the definition of the Lambert W function,

$$(S - A)\tau = W(A_d \tau Q),$$

which can be solved for S:

$$S = \frac{1}{\tau}W(A_d \tau Q) + A. \tag{2.7}$$

Substituting (2.7) into (2.6) yields

$$W(A_d \tau Q)e^{W(A_d \tau Q)+A\tau} = A_d \tau, \tag{2.8}$$

which is a matrix equation in which Q is the only unknown matrix. Together, (2.8) and (2.7) make up a fully-determined system in which Q and S are the two unknown matrices. Since (2.8) does not contain S, we can solve the system as follows:

1. Solve (2.8) for Q using Newton iteration or any other numerical method.
2. Substitute Q into (2.7) and solve for S directly

From (2.4), we can see that the eigenvalues of S are the poles of system (2.3).

Recall that the Lambert W function has an infinite number of branches in the complex plane. Each of these branches produces a different S matrix with its own eigenvalues. Thus, the total solution to system (2.3) can be written

$$x(t) = \sum_{k=-\infty}^{\infty} e^{S_k t} \Gamma_k$$

and the eigenvalues of S_k make up the infinite root spectrum of the system. Yi [24] postulated that for any system without repeated roots at the origin, the principal branch $k = 0$ would produce the rightmost roots of the system, and therefore stability could be determined by examining only the principal branch. This postulate was proved for the scalar case by Shinozaki & Mori [25] by mapping a circle in the complex plane onto the Lambert W function and showing monotonicity with respect to Lambert W branch number. However, this proof cannot be generalized to the matrix case, and such a proof is beyond the scope of this paper.

Because the above method produces the roots of the system directly, it can easily be reversed to select the dominant poles of a controlled system by designing a feedback gain matrix F:

$$\dot{x}(t) = Ax(t) + BFx(t - \tau).$$

In this case, we add an extra set of equations:

$$eigs(S_0) = \vec{\lambda} \tag{2.9}$$

where $\vec{\lambda}$ is the vector of desired dominant poles. Equations (2.7), (2.8), and (2.9) can then be solved together for the principal branch to find S_0, Q_0, and F, which will place the rightmost roots at the desired locations. In general, the roots cannot be placed arbitrarily far to the left for a given system and delay. However, there is no analytical formulation available for the exact limits of the allowable root placement.

CHAPTER 3

ESTIMATOR DESIGN USING THE LAMBERT W METHOD

Estimator Formulation

To solve the problem of determining the un-delayed state $x(t)$ given the delayed measurements y from equation (1.1), we consider a Delay-Compensating Estimator (DCE) of the form:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L[y(t) - \hat{y}(t)]$$

$$\hat{y}(t) = \sum_{i=1}^{p} C_i\hat{x}(t - \tau_i) \tag{3.1}$$

where L is an estimator gain which needs to be designed. This is essentially a state observer in which the estimated output $\hat{y}(t)$ is intentionally delayed by the same amount as the measurement $y(t)$ so that the error $y(t) - \hat{y}(t)$ is taken at the same time step. Thus, the DCE's knowledge of the error is delayed, but the error itself is coherent. A block diagram of the DCE is shown in Figure 3.1.

The induced delay might evoke some similarity to the well-studied Smith predictor [13], [14]. The key difference is that the estimation error feedback gain in the DCE is separate from any controller gain. This gives the DCE two major advantages over the classic Smith predictor. First, it allows the plant to be de-coupled from the estimator, so that the delay compensation process is completely abstracted from the control process. This has the side benefit of allowing the extraction of un-delayed measurement data from an uncontrolled (or externally controlled) plant. Second, it allows the controller and estimator gains to be tuned separately, so that the controller performance can be specified independently from the estimator performance. The main disadvantage of the DCE compared to the Smith predictor is that in addition to the controller gain F, the DCE requires the design of the estimator gain matrix L, which in turn requires a design method suitable for time-delay systems.

The estimator error is defined as

$$e_x(t) = x(t) - \hat{x}(t). \tag{3.2}$$

Taking the time derivative and substituting (1.1) and (3.1) into (3.2) yields

$$\dot{e}_x(t) = Ae_x(t) - L\sum_{i=1}^{p} C_i e_x(t - \tau_i)\,, \tag{3.3}$$

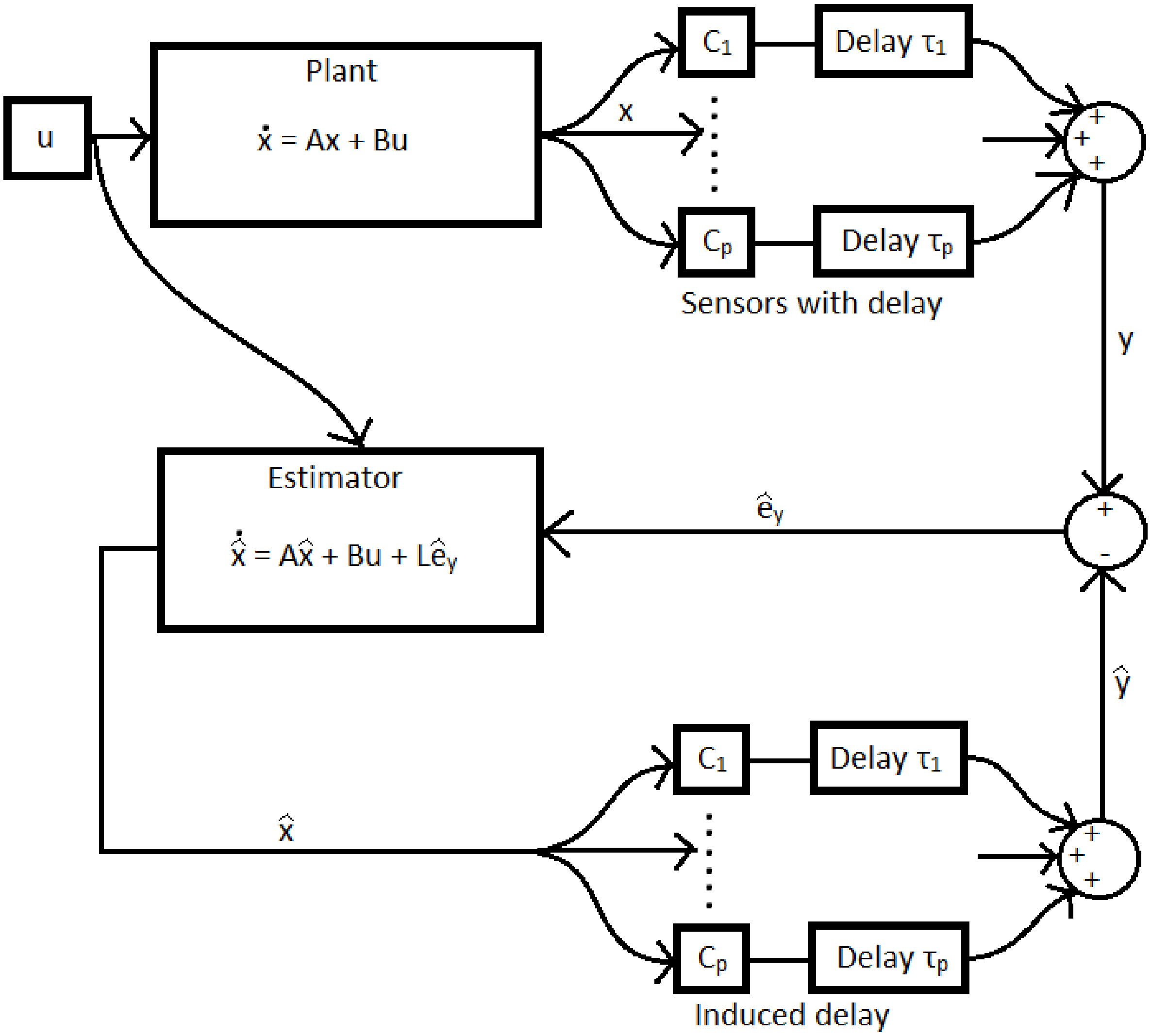

FIGURE 3.1. Block diagram of the proposed Delay-Compensating Estimator (DCE).

which has a similar form to the time delay system solved by the Lambert W method,

$$\dot{x}(t) = Ax(t) + A_d x(t - \tau).$$

The error $e_x(t)$ is what the estimator should drive to zero. Therefore, L needs to be designed according to a method which accounts for time delays. While this may have been an intractable problem when the Smith predictor was first proposed, recent advances in time-delay analysis have already provided solutions, which this work will build upon. In particular we will make use of the Lambert W method of analysis for time delay systems.

Extension of the Lambert W Method to Multiple Delays

The Lambert W method [24] outlined in Section 2 was formulated only for a single delay τ. However, it can be extended to systems with multiple delays by introducing an extra matrix V and associated equation. Starting with the following system with multiple feedback delays, which has the same form as the estimator error dynamics (3.3),

$$\dot{x}(t) = Ax(t) + \sum_{i=1}^{p} A_{d,i} x(t - \tau_i)\,, \tag{3.4}$$

and again assuming a solution of the form (2.4), we get

$$Se^{St}\Gamma - Ae^{St}\Gamma - \sum_{i=1}^{p} A_{d,i} e^{S(t-\tau_i)}\,\Gamma = 0,$$

which implies

$$S - A - \sum_{i=1}^{p} A_{d,i} e^{-S\tau_i} = 0. \tag{3.5}$$

We select a single non-zero delay τ_q and then lump all the remaining delays into a new matrix V_q, defined as

$$V_q = -\sum_{i=1,\ \ i \neq q}^{p} A_{d,i} e^{-S\tau_i} \tag{3.6}$$

which allows us to rewrite (3.5) as

TABLE 1. Summary of Equations for the Lambert W method

Single Delay	Multiple Delays
Analysis $\begin{cases} W_k(-LC\tau Q_k)e^{W_k(-LC\tau Q_k)+A\tau} = -LC\tau \\ S_k = \frac{1}{\tau}W_k(-LC\tau Q_k) + A \end{cases}$ $k = -\infty, \ldots -1, 0, 1, \ldots \infty$ Unknowns: Q_k, S_k	Analysis $\begin{cases} W_k\left(-LC_q\tau_q Q_{q,k}\right)e^{W_k\left(-LC_q\tau_q Q_{q,k}\right)+A\tau_q-V_{q,k}\tau_q} = -LC_q\tau_q \\ S_k = \frac{1}{\tau_q}W_k\left(-LC_q\tau_q Q_{q,k}\right) + A - V_{q,k} \\ V_{q,k} = L\sum_{i=1,\ i\neq q}^{p} C_i e^{-S_k\tau_i} \end{cases}$ $k = -\infty, \ldots -1, 0, 1, \ldots \infty$ Unknowns: $Q_{q,k}, S_k, V_{q,k}$
Design $\begin{cases} W_0(-LC\tau Q_0)e^{W_0(-LC\tau Q_0)+A\tau} = -LC\tau \\ S_0 = \frac{1}{\tau}W_0(-LC\tau Q_0) + A \\ eigs(S_0) = \vec{\lambda} \end{cases}$ Unknowns: Q_0, S_0, L	Design $\begin{cases} W_0\left(-LC_q\tau_q Q_{q,0}\right)e^{W_0\left(-LC_q\tau_q Q_{q,0}\right)+A\tau_q-V_{q,0}\tau_q} = -LC_q\tau_q \\ S_0 = \frac{1}{\tau_q}W_0\left(-LC_q\tau_q Q_{q,0}\right) + A - V_{q,0} \\ V_{q,0} = L\sum_{i=1,\ i\neq q}^{p} C_i e^{-S_0\tau_i} \\ eigs(S_0) = \vec{\lambda} \end{cases}$ Unknowns: $Q_{q,0}, S_0, V_{q,0}, L$

$$S - A + V_q = A_{d,q}e^{-S\tau_q}.$$

Note that the selection of q is arbitrary, so long as $\tau_q \neq 0$. As we will see, the solution matrix S_k is not dependent on q. Following the same procedure as in the single-delay method derivation (see Chapter 2),

$$\tau_q\left(S - A + V_q\right)e^{S\tau_q}e^{-A\tau_q}e^{V_q\tau_q} = A_{d,q}\tau_q e^{-A\tau_q}e^{V_q\tau_q} \tag{3.7}$$

Note that Q_q is now dependent on our selection of τ_q when we defined V_q:

$$\tau_q\big(S - A + V_q\big)e^{(S-A+V_q)\tau_q} = A_{d,q}\tau_q Q_q$$

This can once again be re-written in terms of the matrix Lambert W function:

$$W\big(A_{d,q}\tau_q Q_q\big) = \tau_q\big(S - A + V_q\big) \tag{3.8}$$

Solving for S and plugging back into (3.7), we end up with the following set of equations to solve:

$$S = \frac{1}{\tau_q} W\big(A_{d,q}\tau_q Q_q\big) + A - V_q \tag{3.9}$$

$$W\big(A_{d,q}\tau_q Q_q\big)e^{W(A_{d,q}\tau_q Q_q)+A\tau_q - V_q\tau_q} = A_{d,q}\tau_q \tag{3.10}$$

For any $q \in [1, 2, \dots p] \mid \tau_q \neq 0$, (3.6), (3.9), and (3.10) form a fully-determined system in which S is independent of the q selected. Note that in the case of a single delay, $p = 1$, $V_q = 0$, and the equations (3.9) and (3.10) reduce to (2.7) and (2.8).

As was the case for a single delay, we can transform the solution method into a design method by replacing $A_{d,q}$ with the feedback control matrix BF and solving equations (3.6), (3.9), (3.10), and (2.9) with F as an unknown matrix.

Application of the Lambert W Pole Placement Method to Delay-Compensating Estimator

The Lambert W method is easily adapted to the DCE. We only need to replace the matrix A_d with the product $-LC$ in the single-delay case, or

$$A_{d,q} \to -LC_q$$

$$V_q = L\sum_{i=1,\ i\neq q}^{p} C_i e^{-S\tau_i}$$

in the multiple-delay case. These equations are summarized in Table 3.1.

In the single-delay analysis case, the equations can be solved sequentially – first for Q_k, then for S_k. However, for the other three cases, the equations must be solved simultaneously.

The analysis equations together form a fully-determined system of three $n \times n$ matrix equations and three $n \times n$ unknown matrices S, Q, V. However, the design equations are under-determined, with three $n \times n$ matrix equations and up to n scalar eigenvalue assignment equations, but three $n \times n$ unknown matrices S, Q, V and one $m \times n$ unknown matrix L. Thus there are $(m \times n) - n$ degrees of freedom, and the design solution is not unique.

Not all of the eigenvalues of the principal branch need to be specified when designing the gain L. In fact, the fewer eigenvalues are specified, the farther left they can be placed (a limitation that will be discussed in the next section). Assigning fewer eigenvalues increases the number of degrees of freedom in the under-determined system, which may allow for increased radius of convergence. However, if assigning fewer than n eigenvalues, one must be careful to compare the assigned eigenvalues to the *rightmost* eigenvalues of the calculated S_0.

Limitations of the Method

The Lambert W method is a powerful tool for theoretical analysis of time-delay systems due to the analytic solutions it produces. However, the method does have some limitations, some of which were outlined by Yi [24].

The major limitation is that for a given system with given delays, the poles cannot be placed arbitrarily far to the left on the complex plane. In general, as the rightmost poles in the principal S_0 branch are pushed further left, the extra poles from the non-principal $S_{k \neq 0}$ branches move to the right, until at a certain point, they switch places with the principal branch. This creates a discontinuity in the gradient of the poles with respect to the gain matrix, which causes numerical solvers based on gradient descent (such as the Levenberg-Marquardt algorithm used by `fsolve` in MATLAB [26]) to fail to converge. This was noted by Yi, and was also observed for the multiple-delay case developed here (see the example later in the chapter). The roots could theoretically be optimized to be as far left as possible using non-smooth optimization as proposed by Michiels, et al [6], but such an approach was not pursued here.

Another limitation is that of observability for the DCE. While no theoretical reason has been developed, a requirement for successful placement of the poles of the DCE has been noticed experimentally, namely:

$$rank\left(\sum_i C_i\right) = n. \tag{3.11}$$

The observability condition of (3.11) is a huge limitation on the number of systems that can be estimated with the proposed method. Not all states are measurable for most systems, which leaves $\sum_i C_i$ rank-deficient and makes those systems un-observable to the DCE, even if they are observable in the classical sense. Therefore, the DCE cannot function as a general replacement for the Luenberger observer, although it may be possible to use both in tandem to reconstruct the state vector and then filter out the effect of the delays. This would be a topic for future research.

Observability for single-delay systems was investigated in [27], but the result was derived for a slightly different system, and the requirements for observability were less stringent than that of (3.11). Specifically, for the system

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

$$y(t) = Cx(t),$$

the condition for observability was found to be that the matrix

$$C(sI - A - A_d e^{-s\tau})^{-1}$$

have all linearly independent columns. This result does not appear to carry over to the design of the DCE.

Considerations for Numerical Solution of the Method

Because a numerical solution is required for the equations in Table 3.1, initial guesses, solver algorithms, and convergence properties are important considerations for application of the Lambert W method.

There have been some recent studies into the numerical properties of the single-delay case. Convergence of the numerical solution for this case is not guaranteed for arbitrary initial conditions. In fact, as was shown in [28], depending on the initial guess for Q, several different sets of roots can be found using the same branch of the Lambert W function. This suggests that, in general, Q is not unique for a particular branch. Ivanoviene & Rimas [29] showed that the numerical reliability of the method was improved by solving for $D_k = W_k(A_d \tau Q_k)$ as a lumped quantity instead of just for Q_k. It was shown that which branch the solver converged to was dependent upon the initial guess for D_k, and that an initial guess of $D_k^{(init)} = W_k(-A_d \tau e^{A\tau})$ would converge to the k-th branch of the Lambert W function.

The numerical improvement from [29] yields consistent results because in the single-delay case, D_k is solved first, and then the solution S_k is found as an explicit function of D_k. Thus, the only initial guess needed is for D_k. This is not directly applicable to the multi-delay case, because all three unknown matrices $S_k, Q_{q,k}, V_{q,k}$ need to be solved for simultaneously. Numerical reliability can be improved for the principal branch by solving for the lumped quantity $D_{q,k} = W_k(A_{d,q} \tau_q Q_{q,k})$ instead of $Q_{q,k}$, but the branch that is converged to cannot be so easily controlled by a single initial matrix, because initial guesses are needed for both $S_k^{(init)}$ and $D_{q,k}^{(init)}$. A separate guess is not strictly needed for $V_{q,k}^{(init)}$, since it can be expressed as an explicit function of $S_k^{(init)}$ via (3.6).

An initial guess for $S_k^{(init)} = A + \sum_i A_{d,i}$ is fairly straightforward for the principal branch. The solution to an un-delayed linear system $\dot{x}(t) = Ax(t) + A_d x(t)$ is $x(t) = e^{(A+A_d)t} x(0)$ so it is apparent that $A + \sum_i A_{d,i}$ would be the solution matrix S in equation (2.4) if the delay was zero. However, initial guesses for the non-principal branches are much more difficult to find, as they require some *a priori* knowledge of the solution S_k for that particular branch. An initial guess of $D_{q,k}^{(init)} = \tau_q\left(S_k^{(init)} - A + V_{q,k}^{(init)}\right)$ generally provides a good starting point which agrees with the initial guess for $S_k^{(init)}$. However, this means the convergence of the algorithm hinges solely on the selection of $S_k^{(init)}$, rather than $D_{q,k}^{(init)}$ for the single-delay case. For non-principal branches, this initial guess selection is still an open problem.

Several numerical methods were tested, but the most consistent and reliable algorithm for the design case seemed to be the Levenberg-Marquardt algorithm for `fsolve` in MATLAB (in fact, this is the only algorithm for `fsolve` which can handle non-square systems). For the analysis case, the trust-region dogleg algorithm was usable because the system is square, and it out-performed Levenberg-Marquardt in terms of convergence speed. More sophisticated numerical methods might offer better performance, but in the interest of ease of implementation, these were not pursued.

Example: Simple Harmonic Oscillator

Consider the simple mass-spring-damper system of Figure 3.2, with the following parameter values: $m = 5,\ k = 3,\ c = 2.2$ (in consistent units). The dynamics for this system are

$$m\ddot{z} + c\dot{z} + kz = u$$

which can be written in linearized form as:

$$\dot{x} = Ax + Bu$$

$$x = \begin{Bmatrix} z \\ \dot{z} \end{Bmatrix}, \qquad A = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.6 & -0.44 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & 0 \\ 0 & 1/m \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0.2 \end{bmatrix}.$$

The eigenvalues of this system are $\lambda = -0.22 \pm 0.7427i$.

Suppose we have a sensor which detects the mass position z with a delay of $\tau_1 = 0.1$ seconds, and another sensor which detects the velocity $\dot{z}$ with a slightly longer delay of $\tau_2 = 0.13$ seconds. Then the measured output y is

$$y(t) = C_1 x(t - \tau_1) + C_2 x(t - \tau_2)$$

$$C_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad C_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Since this system has the form of (1.1), we can use the extended Lambert W method to design a DCE which will predict the current state $x(t)$ despite the measurement delays.

For now, we select estimator eigenvalues of $\lambda_{DCE} = \{-3, -3.1\}$. Then, solving the Multiple Delay Design equations from Table 3.1 using `fsolve` in MATLAB produces an estimator gain matrix of

$$L = \begin{bmatrix} 2.1511 & 0.8435 \\ -0.5007 & 1.8644 \end{bmatrix}.$$

The initial guess for $L^{(init)}$ was found by using the `place` command in MATLAB, which calculates a gain matrix which places the poles at the desired locations for an un-delayed system. The initial guess used for $S_0^{(init)}$ was $S_0^{(init)} = A - L^{(init)} \sum C_i$, and the initial guess for $D_{1,0}^{(init)}$ was $D_{1,0}^{(init)} = \tau_1 \left(S_0^{(init)} - A + V_{1,0} \right)$ where $V_{1,0}$ was found from (3.10). The simulated results of this system design are shown in Figure 3.2. The estimator error quickly decays to zero with no oscillation, which is characteristic of an overdamped system and exactly what we would expect from dominant eigenvalues of $\{-3, -3.1\}$.

Alternate Cases Showing Convergence Issues

Now suppose the velocity measurements for the above example were unavailable, and we only had the position measurements, i.e. $y(t) = C_1 x(t - \tau_1)$. When we try to apply the Lambert W design method, the numerical solver fails to converge to a solution. This case breaks condition (3.13) since $rank(C_1) = 1$, but $n = 2$ for the system.

However, the measurement matrices C_i need not sum to identity. For example the following set of measurements (picked somewhat arbitrarily, without a particular physical meaning in mind) *does* converge to a solution:

$$C_1 = \begin{bmatrix} 1 & -2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \qquad C_2 = \begin{bmatrix} 0 & 0 \\ 1.4 & 0 \\ 0 & 0 \end{bmatrix}, \qquad C_3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 2 & 0 \end{bmatrix}$$

$$\tau_1 = 0.1, \qquad \tau_2 = 0.13, \qquad \tau_3 = 0.22$$

converges to

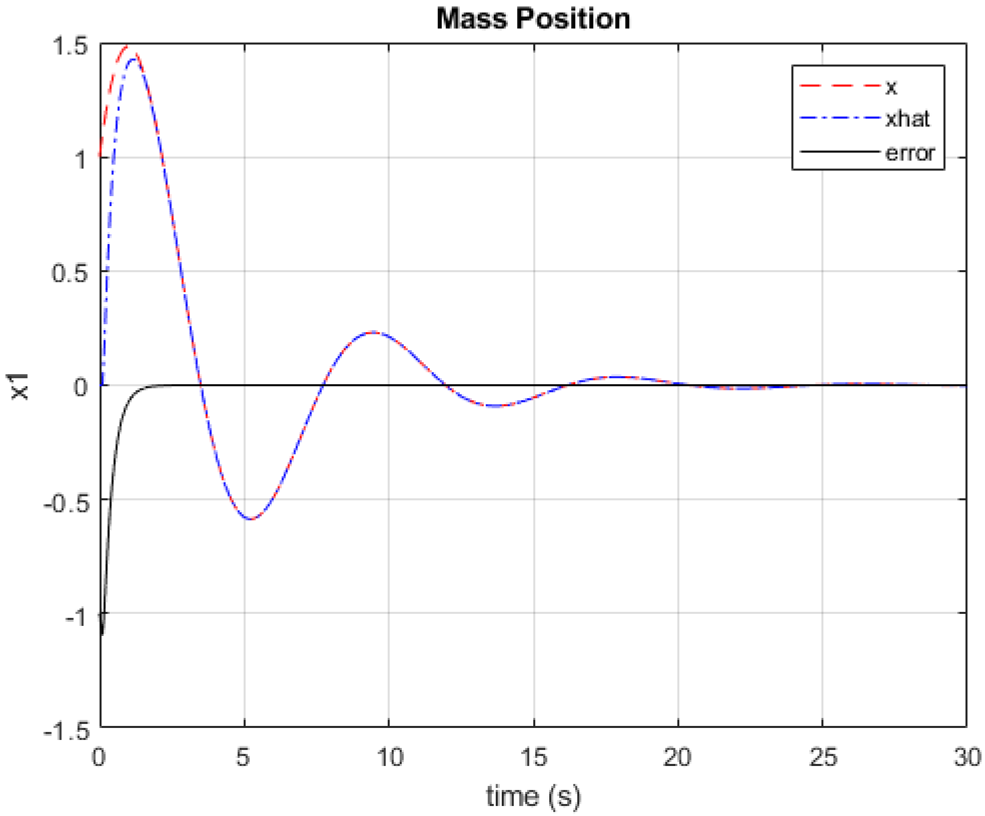

FIGURE 3.2. Simulated DCE performance for a mass-spring-damper system.

$$L = \begin{bmatrix} -0.3651 & 0.1633 & 0.8244 \\ -0.9212 & 0.0548 & 0.1830 \end{bmatrix}.$$

To illustrate the limits on how far left the rightmost eigenvalues can be placed, consider attempting to place the estimator eigenvalues at $\lambda_{DCE} = \{-30, -30.1\}$ instead of $\{-3, -3.1\}$. The solver fails to converge in this case because the eigenvalues simply cannot be placed that far left for the given system with the given delays.

Conclusion

The Lambert W method of analysis for time-delay systems was extended to the multiple-delay case by adding an additional matrix $V_{q,k}$ and associated equation, which is explicit in S_k. Analytically, the result is equivalent to the original method, but with the capability to handle multiple delays. Numerically, however, the new method requires twice as many initial guesses as the single-delay method. Reliable initial guesses have only been found for the principal branch of the multiple-delay method, which is sufficient for the pole-placement design process, but not for a full analysis of the entire root spectrum. The formulation of initial guesses for non-principal branches remains an unsolved problem.

CHAPTER 4

EFFECTS OF DELAY UNCERTAINTY ON ESTIMATOR DESIGN

In the previous chapter, knowledge of the nominal delays was required to design the gain matrix L for the DCE. In general, however, there are errors and uncertainties in the modeling of the delay, which can cause additional estimation errors. This chapter deals with analysis of the DCE under these conditions, as well as development of acceptable bounds on the delays for a given estimator design.

Formulation

We consider the system in Figure 4.1, in which the real measurement delay τ_r is not necessarily equal to the induced delay τ_d that the system was designed around. The dynamics of the DCE for this system are

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L\sum_{i=1}^{p} C_i\left[x\left(t-\tau_{r,i}\right) - \hat{x}\left(t-\tau_{d,i}\right)\right] \tag{4.1}$$

and the true estimator state error is

$$e_x(t) = x(t) - \hat{x}(t).$$

Taking the time derivative and substituting in (4.1) and (1.1),

$$\dot{e}_x(t) = Ax(t) + \cancel{Bu(t)} - A\hat{x}(t) - \cancel{Bu(t)} - L\sum_{i=1}^{p} C_i\left[x\left(t-\tau_{r,i}\right) - \hat{x}\left(t-\tau_{d,i}\right)\right]$$

$$\dot{e}_x(t) = Ae_x(t) - L\sum_{i=1}^{p} C_i\left[x\left(t-\tau_{r,i}\right) - \hat{x}\left(t-\tau_{d,i}\right)\right].$$

By adding and subtracting $x\left(t-\tau_{d,i}\right)$ inside the brackets and re-grouping, this can be rewritten as

$$\dot{e}_x(t) = Ae_x(t) - L\sum_{i=1}^{p} C_i e_x\left(t-\tau_{d,i}\right) - L\sum_{i=1}^{p} C_i\left[x\left(t-\tau_{r,i}\right) - x\left(t-\tau_{d,i}\right)\right]. \tag{4.2}$$

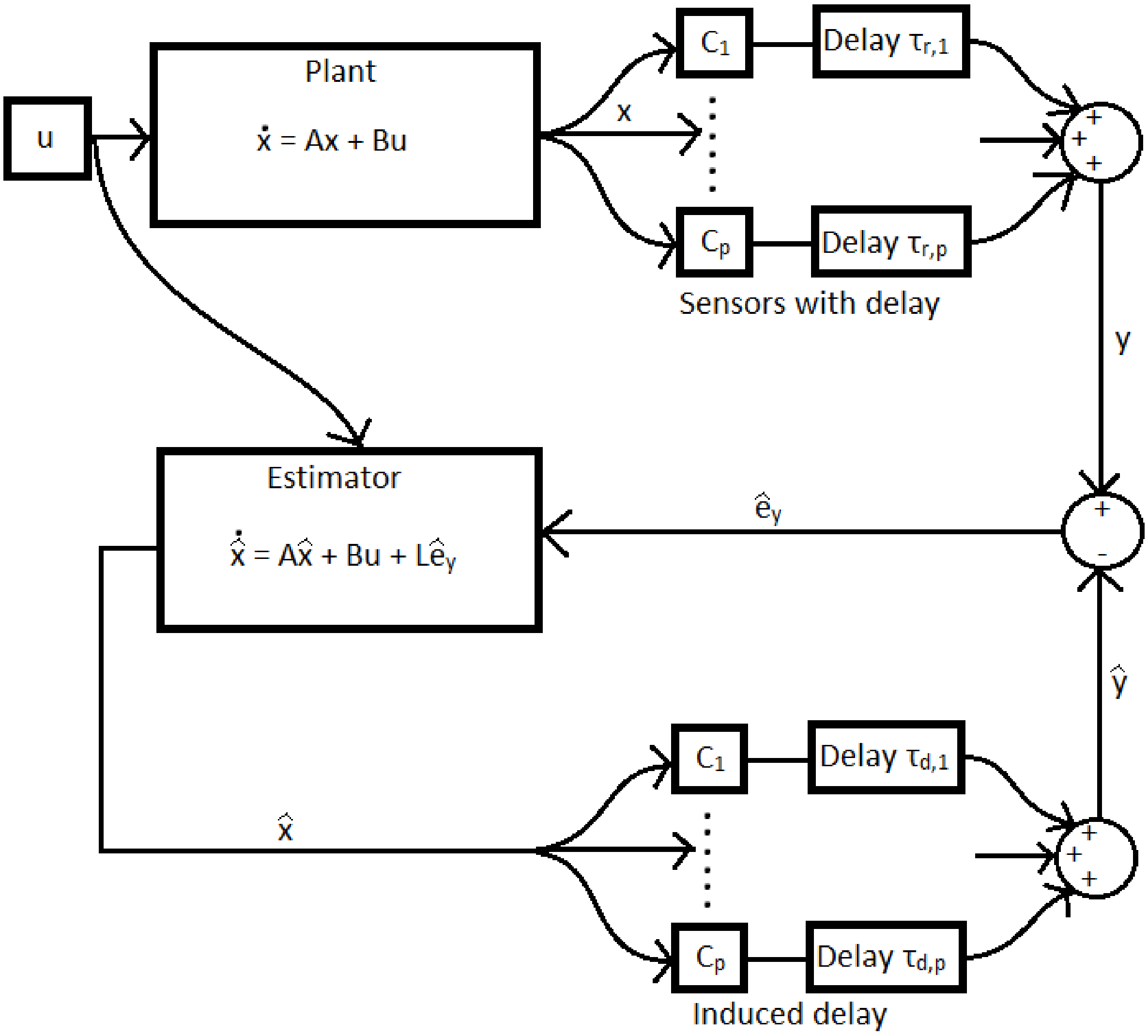

FIGURE 4.1. Block diagram of the estimator with uncertainty in delay modeling.

The term e_x is the true error between the real state x and the estimated state $\hat{x}$ at any given time. Taking the Laplace transform of (4.2) reveals two independent responses for $e_x(t)$, one dependent on the initial error, and one dependent on the delay error:

$$\begin{aligned}
&E_x(s) = \Big(sI - A \\
&\boxed{\text{Delay error term} \rightarrow} \; + L\sum_{i=1}^{p} C_i \exp\left(-\tau_{d,i}s\right)\Big)^{-1}\Big\{-L\sum_{i=1}^{p} C_i\left[\exp\left(-\tau_{r,i}s\right) - \exp\left(-\tau_{d,i}s\right)\right]\Big\}X(s) \\
&\boxed{\text{Initial error term} \rightarrow} \; + \Big(sI - A + L\sum_{i=1}^{p} C_i \exp\left(-\tau_{d,i}s\right)\Big)^{-1} e_x(0) \qquad (4.3)
\end{aligned}$$

The initial error portion of (4.3) is exactly the same as the response of the original designed estimator, which means the initial error decay is unaffected by uncertainty in the delays. Because the system is linear, it can be expressed as a superposition of the designed response and an error due to uncertainty in the delay modeling. The delay-uncertainty term can then be analyzed separate from the designed response.

First, we define the error in the output domain:

$$e_y(t) = \sum_{i=1}^{p} C_i e_x\left(t - \tau_{d,i}\right) \qquad (4.4)$$

This can be interpreted as a transformation of the true error from the state domain to the delayed output domain. Another interpretation is that $e_y(t)$ is the input that the DCE was initially designed to see.

It is also useful to define the output error that is actually seen by the estimator, which we will call the *estimated error*:

$$\hat{e}_y(t) = y(t) - \hat{y}(t) = \sum_{i=1}^{p} C_i\left[x\left(t - \tau_{r,i}\right) - \hat{x}\left(t - \tau_{d,i}\right)\right]. \qquad (4.5)$$

This quantity is important because it is what the estimator attempts to drive to zero. It is also worth noting that when the delay is perfectly modeled, $\{\tau_r\} = \{\tau_d\}$ and $\hat{e}_y(t) = e_y(t)$.

The estimated error $\hat{e}_y(t)$ has the dynamics of the original designed DCE. As discussed previously, the linearity of the system allows us to separate the total error into the designed error response and the delay-uncertainty error. We define the difference between the true error and the estimated error as the residual error $\bar{e}_y(t)$:

$$\bar{e}_y(t) = e_y(t) - \hat{e}_y(t) \tag{4.6}$$

Plugging (4.4) and (4.5) into (4.6), we find that

$$\bar{e}_y(t) = \sum_{i=1}^{p} C_i\left[x\left(t - \tau_{d,i}\right) - x\left(t - \tau_{r,i}\right)\right]. \tag{4.7}$$

The residual error represents the portion of the error that is due to delay uncertainty.

By design, the DCE drives the estimated error $\hat{e}_y(t)$ to zero at a rate determined by the selected poles. Ideally, this rate is much faster than the estimatee (that is, the dynamic system being estimated) settles to its final value. This leaves a period of time after the initial estimator error has decayed, but before the estimatee has settled, where the residual error is the dominant portion of the total error. We will call this the *quasi-steady-state* condition for the estimator, and we can think of the residual error as also being the quasi-steady-state error.

Yet another error quantity that is useful to define is the measurement error due to delay,

$$\tilde{y}(t) = \sum_{i=1}^{p} C_i\left[x(t) - x\left(t - \tau_{r,i}\right)\right]. \tag{4.8}$$

This is also the quasi-steady-state error we would have if we did not model the measurement delay in the estimator feedback at all – that is, if we used a classic Luenberger observer instead.

Evaluation of Estimator Performance

Before bounds on the acceptable delay can be developed, we first need a metric with which to evaluate the performance of the estimator. Li & Zhao [30] proposed a Measurement Error Reduction Factor (MERF), which quantifies the improvement that the estimator makes over taking the (erroneous) measurements at face value. MERF is the ratio of the estimator error to the measurement error, and is defined for a discrete system as:

$$MERF = \left(\frac{\sum_{i=1}^{M}[\left(y_{true}(i) - \hat{y}(i)\right)'\left(y_{true}(i) - \hat{y}(i)\right)]}{\sum_{i=1}^{M}[\left(y_{true}(i) - y_{meas}(i)\right)'\left(y_{true}(i) - y_{meas}(i)\right)]}\right)^{\frac{1}{2}}$$

The smaller the MERF value, the better the estimator performance. A MERF less than 1 means the estimated state is more accurate than the measurement, while a MERF greater than 1 means the estimator provides a worse estimate of the system than just using the measurement. This provides a simple method of determining acceptable bounds for the estimator: the estimator is considered failed if MERF > 1.

Note that, in practice, MERF is typically calculated separately for states with different units. That is, MERF for cartesian positions will be calculated separately from MERF for velocity, both of which will be separate from MERF for angles, and so on.

For the DCE, the estimator error is simply $e_y(t)$, and the measurement error is $\tilde{y}(t)$. MERF therefore tells how much of an improvement the induced delay in the estimator feedback loop makes compared to a Luenberger observer. However, because $e_y(t)$ is dependent on the unavoidable and essentially arbitrary initial error, but $\tilde{y}(t)$ is not, it is more useful to consider only the residual error $\bar{e}_y(t)$ due to errors in delay modeling, (which is also the quasi-steady-state error). Writing MERF using these terms and in an integral form for continuous systems yields the following expression:

$$MERF = \left(\frac{\int_0^\infty[\bar{e}_y(t)'\bar{e}_y(t)]dt}{\int_0^\infty[\tilde{y}(t)'\tilde{y}(t)]dt}\right)^{\frac{1}{2}} \tag{4.9}$$

The limiting point at which the estimator is considered failed is when $MERF = 1$. In this case, the square root is irrelevant, and we can write

$$\int_0^\infty \left[\bar{e}_y(t)'\bar{e}_y(t)\right]dt = \int_0^\infty [\tilde{y}(t)'\tilde{y}(t)]dt.$$

Substituting (4.7) and (4.8) into this expression yields

$$\int_0^\infty \left\{\sum\nolimits_i C_i\left[x\left(t-\tau_{d,i}\right)-x\left(t-\tau_{r,i}\right)\right]\right\}' \left\{\sum\nolimits_i C_i\left[x\left(t-\tau_{d,i}\right)-x\left(t-\tau_{r,i}\right)\right]\right\} dt$$

$$= \int_0^\infty \left\{\sum\nolimits_i C_i\left[x(t)-x\left(t-\tau_{r,i}\right)\right]\right\}' \left\{\sum\nolimits_i C_i\left[x(t)-x\left(t-\tau_{r,i}\right)\right]\right\} dt \quad (4.10)$$

Note that (4.10) is written entirely in terms of the system being estimated and the delays. This means the radius of acceptable delays is independent of the estimator gain L. The robustness of the DCE to delay uncertainty when used for pure observation is therefore not dependent on the estimator design, but only on the estimatee and the delay uncertainty. This occurs because MERF only quantifies the measurement improvement (or lack thereof) due to the induced delay in the estimator feedback, so any occurrences of the estimator gain L are cancelled out.

Delay Bounds for Acceptable Performance

Now that we have a measure of performance for the DCE, we will use it to develop bounds on the delay for acceptable performance.

Scalar System with a Single Delay

In the special case when $A = a$ is scalar and there is only one delay ($p = 1$), we can get an explicit, closed-form solution for the critical delay $\tau_{r_{crit}}$ at which the DCE fails. Note that if the integrands on both sides of (4.10) are equal, then the integrals must also be equal. We can then write (4.10) as

$$e^{2at}\left[e^{-a\tau_d}-e^{-a\tau_r}\right]^2 x_0^2 = e^{2at}\left[1-e^{-a\tau_r}\right]^2 x_0^2$$

which expands to

$$e^{-2a\tau_d}-2e^{-a\tau_d}e^{-a\tau_r}+e^{-2a\tau_r} = 1-2e^{-a\tau_r}+e^{-2a\tau_r}.$$

Isolating for $e^{-a\tau_r}$,

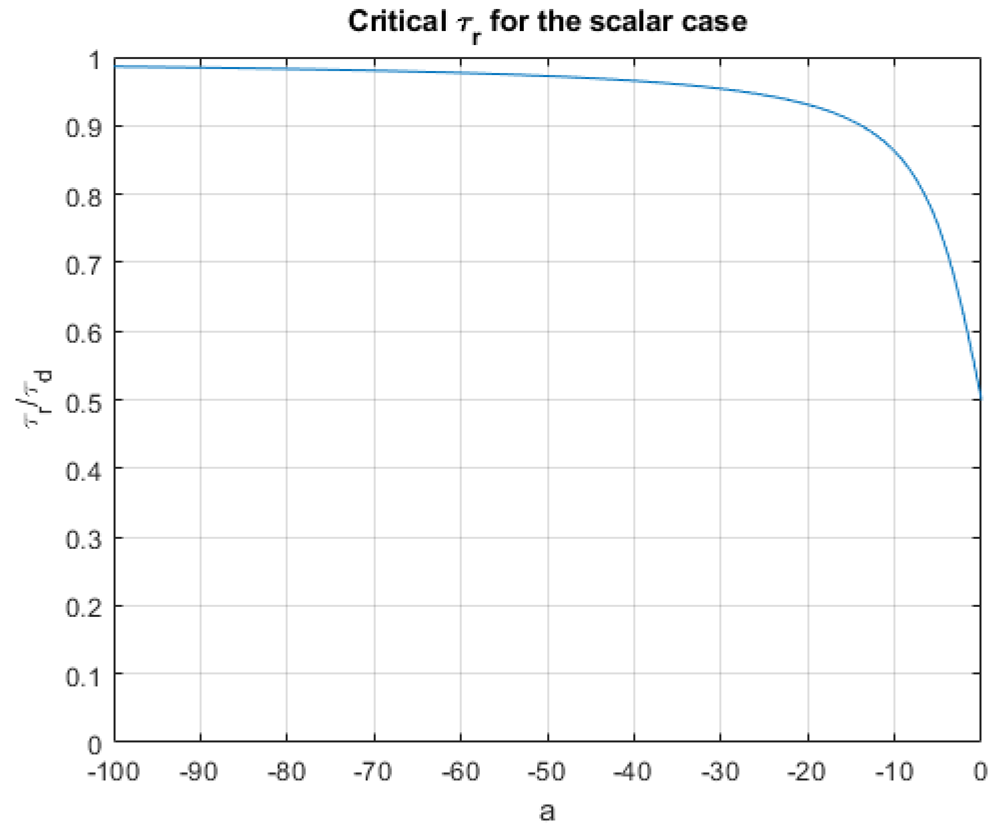

FIGURE 4.2. Critical values of τ_r below which the estimator is considered failed.

$$e^{-a\tau_r} = \frac{1-e^{-2a\tau_d}}{2-2e^{-a\tau_d}} = \frac{1}{2}\frac{(1+e^{-a\tau_d})(\cancel{1-e^{-a\tau_d}})}{\cancel{1-e^{-a\tau_d}}}$$

and taking the natural log of both sides to solve for τ_r, we get

$$\tau_{r_{crit}} = \frac{\ln\left[\frac{1}{2}(1+e^{-a\tau_d})\right]}{-a}. \tag{4.15}$$

Eq. (4.15) is plotted in Figure 4.2 and shows that as the system response becomes faster (a becomes more negative), $\tau_{r_{crit}}$ approaches τ_d. Meanwhile, as the response approaches a ramp (a goes to zero), $\tau_{r_{crit}}$ approaches $\tau_d/2$. If a has no imaginary part, then MERF< 1 for all $\tau_r > \tau_d$.

Second-Order system with Pure Imaginary Roots

Interestingly, taking the real part of (4.15) yields a valid $\tau_{r_{crit}}$ for systems of size $n = 2$ if the eigenvalues of A are a purely imaginary conjugate pair (i.e. $x(t) = e^{i\omega t} - e^{-i\omega t}$). In this case, it can be shown that $\tau_{r_{crit}} = {}^{\tau_d}/_2$ is always true. Noting that for an arbitrary complex number z,

$$\ln(z) = \ln|z| + i\varphi$$

$$\varphi = \tan^{-1}\left(\frac{\mathrm{Im(z)}}{\mathrm{Re(z)}}\right)$$

where φ is the complex phase angle, and also making note of Euler's formula,

$$e^{i\omega} = \cos(\omega) + i\sin(\omega),$$

we can express (4.15) as

$$\tau_{r_{crit}} = Re\left\{\frac{\ln\left[\frac{1}{2}\left(1 + e^{-i\omega\tau_d}\right)\right]}{-i\omega}\right\} = Re\left\{\frac{\ln\left[\frac{1}{2}\left(1 + \cos(\omega\tau_d) - i\sin(\omega\tau_d)\right)\right]}{-i\omega}\right\} \tag{4.16}$$

The real part of (4.16) is just the imaginary part of the logarithm divided by $-i\omega$. Thus,

$$\tau_{r_{crit}} = \frac{i\tan^{-1}\left(\frac{-\sin(\omega\tau_d)}{\cos(\omega\tau_d) + 1}\right)}{-i\omega} \tag{4.17}$$

Using the following double-angle trigonometry identities

$$\sin(\omega\tau_d) = 2\sin\left(\frac{\omega\tau_d}{2}\right)\cos\left(\frac{\omega\tau_d}{2}\right)$$

$$\cos(\omega\tau_d) = 2\cos^2\left(\frac{\omega\tau_d}{2}\right) - 1\,,$$

we can rewrite (4.17) as

$$\tau_{r_{crit}} = \frac{i\tan^{-1}\left(\frac{-\sin\left(\frac{\omega\tau_d}{2}\right)}{\cos\left(\frac{\omega\tau_d}{2}\right)}\right)}{-i\omega} = \frac{-i\left(\omega\frac{\tau_d}{2}\right)}{-i\omega}$$

$$\tau_{r_{crit}} = \frac{\tau_d}{2} \tag{4.18}$$

By (4.18), for any system with a purely sinusoidal response, $\tau_r > \frac{\tau_d}{2}$ is the condition for the delayed estimator to be more effective than a standard estimator. Because of the oscillatory nature of a system with pure imaginary roots, as τ_r increases past τ_d, MERF eventually crosses back above 1 as τ_r approaches the period of oscillation. To understand why, consider the limiting case in which $\tau_r = \frac{2\pi}{\omega}$, the period of the oscillation. In this case, the apparent measurement error is zero because measurements are delayed by exactly one oscillation period, which means MERF spikes to infinity. Because of the simplifying assumptions made in deriving (4.18), the equation does not capture this effect.

General *n*-Dimensional System with Multiple Delays

We now return to the general matrix case with multiple delays. Assuming the system's dynamic response is $x(t) = e^{At}x_0$, (4.10) becomes

$$\int_0^\infty \left\{\sum_i C_i\left[e^{A(t-\tau_{d,i})} - e^{A(t-\tau_{r,i})}\right]x_0\right\}' \left\{\sum_i C_i\left[e^{A(t-\tau_{d,i})} - e^{A(t-\tau_{r,i})}\right]x_0\right\} dt$$

$$= \int_0^\infty \left\{\sum_i C_i\left[e^{At} - e^{A(t-\tau_{r,i})}\right]x_0\right\}' \left\{\sum_i C_i\left[e^{At} - e^{A(t-\tau_{r,i})}\right]x_0\right\} dt$$

Combining the summations produces

$$\int_0^\infty x_0' \left[\sum_{i=1}^p \sum_{j=1}^p \left(e^{A(t-\tau_{d,i})} - e^{A(t-\tau_{r,i})}\right)' C_i' C_j \left(e^{A(t-\tau_{d,j})} - e^{A(t-\tau_{r,j})}\right)\right] x_0 dt$$

$$= \int_0^\infty x_0' \left[\sum_{i=1}^p \sum_{j=1}^p \left(e^{At} - e^{A(t-\tau_{r,i})}\right)' C_i' C_j \left(e^{At} - e^{A(t-\tau_{r,j})}\right)\right] x_0 dt$$

Next, move the integral inside the summation and isolate the time-dependent portions

$$x_0' \left[\sum_i \sum_j (e^{-A\tau_{d,i}} - e^{-A\tau_{r,i}})' \int_0^\infty (e^{At})' C_i' C_j e^{At} dt \, (e^{-A\tau_{d,j}} - e^{-A\tau_{r,j}})\right] x_0$$

$$= x_0' \left[\sum_i \sum_j (I - e^{-A\tau_{r,i}})' \int_0^\infty (e^{At})' C_i' C_j e^{At} dt \, (I - e^{-A\tau_{r,j}})\right] x_0 \quad (4.11)$$

To simplify notation, define

$$\Psi_{ij} = \int_0^\infty (e^{At})' C_i' C_j e^{At} dt \tag{4.12}$$

As long as all eigenvalues of A lie in the left half of the complex plane, this integral can be solved via integration by parts (see Appendix C), yielding

$$A'\Psi_{ij} + \Psi_{ij}A + C_i'C_j = 0 \tag{4.13}$$

Thus, the bound at which the DCE fails can then be expressed as

$$x_0' \left[\sum_i \sum_j (e^{-A\tau_{d,i}} - e^{-A\tau_{r,i}})' \Psi_{ij} (e^{-A\tau_{d,j}} - e^{-A\tau_{r,j}}) \right] x_0$$

$$= x_0' \left[\sum_i \sum_j (I - e^{-A\tau_{r,i}})' \Psi_{ij} (I - e^{-A\tau_{r,j}}) \right] x_0. \tag{4.14}$$

When there are multiple delays ($p > 1$), this equation describes a surface in p-dimensional space, and it must be solved numerically. We can therefore only check whether a given set of delays will cause the estimator to fail or not. If the left side of (4.14) is less than the right side, then MERF is less than 1, and the estimate provided by the DCE is an improvement over the delayed measurement. Conversely, if the left side of (4.14) is greater than the right side, then MERF is greater than 1, and the estimator will fail.

Example: Harmonic Oscillator with Uncertain Delays

Let us return to the harmonic oscillator example from Chapter 3 to show what happens as the real delay changes, while the design delay remains fixed. As a reminder, the system dynamics are

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

$$A = \begin{bmatrix} 0 & 1 \\ -0.6 & -0.44 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 & 0 \\ 0 & 0.2 \end{bmatrix} \tag{4.15}$$

with poles $\lambda = \{-0.22 \pm 0.7746i\}$, and the observer dynamics are

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L\sum_i C_i\left[x\left(t-\tau_{r,i}\right) - \hat{x}\left(t-\tau_{d,i}\right)\right]$$

$$C_1 = \begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}, \qquad C_2 = \begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}, \qquad L = \begin{bmatrix}2.1511 & 0.8435\\-0.5007 & 1.8644\end{bmatrix}$$

$$\tau_{d,1} = 0.1\ s, \qquad \tau_{d,2} = 0.13\ s.$$

When $\{\tau_r\} = \{\tau_d\}$, the residual error $\bar{e}_y$ is 0, so MERF, by the definition in (4.9), is also 0 (i.e. the DCE perfectly matches the true state once it reaches quasi-steady-state). However, suppose there is an error in the delay modeling, and the actual measurement delays are

$$\tau_{r,1} = 0.6, \qquad \tau_{r,2} = 0.2.$$

The resulting estimator output is shown in Figure 4.3. A slight wobble is apparent in the quasi-steady-state estimator error (solid black line). Numerically integrating the estimator and

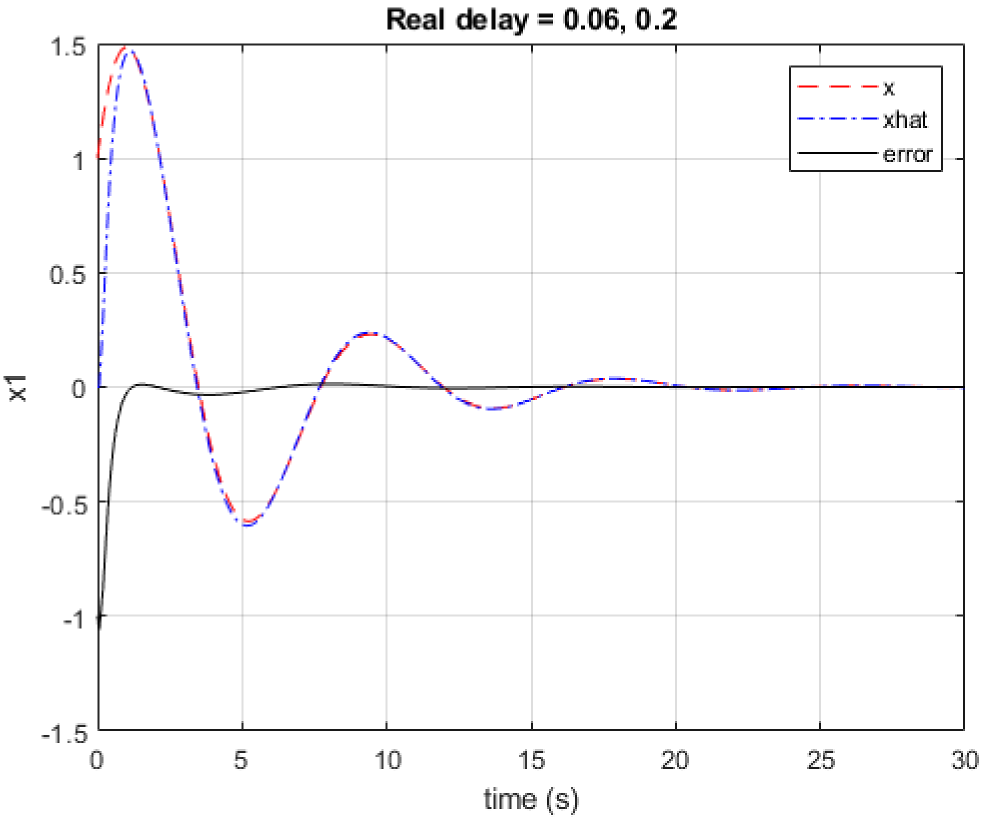

FIGURE 4.3. DCE performance with delay modeling errors.

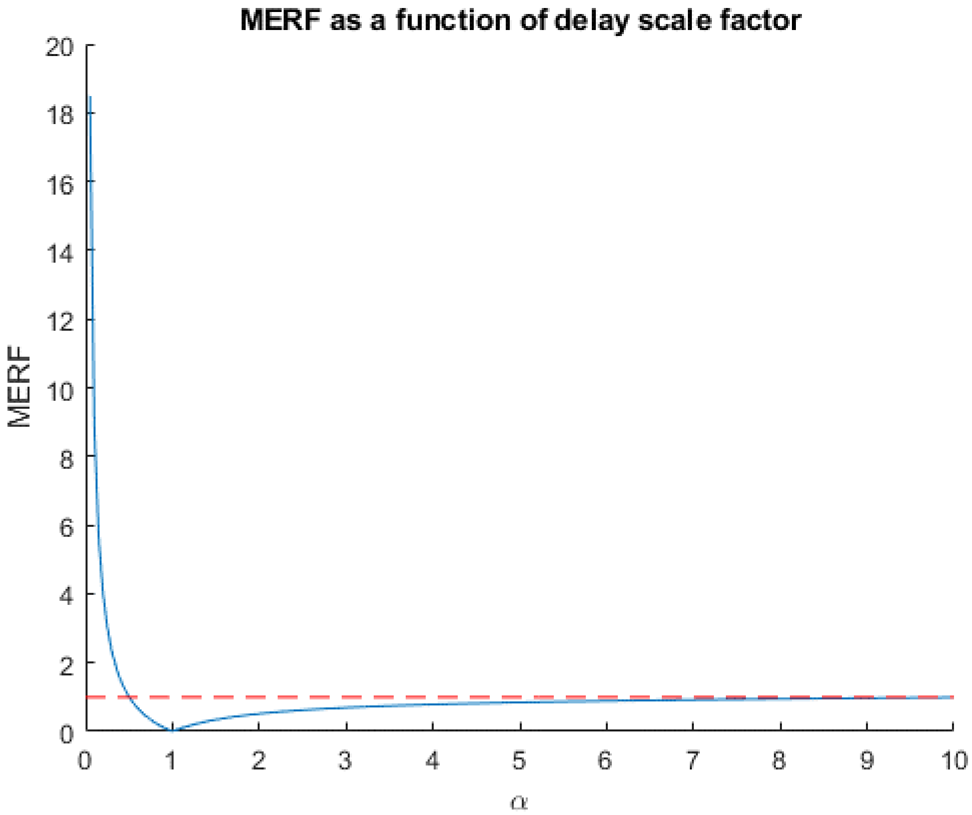

FIGURE 4.4. MERF for the damped oscillator (4.15).

measurement errors (starting after 6 seconds to allow the DCE to reach quasi-steady-state) yields MERF values of 0.7413 for position and 0.3281 for velocity.

To take a more systematic approach, suppose the true delays $\{\tau_r\}$ are a scale factor α of the design delays, that is

$$\tau_{r,i} = \alpha \cdot \tau_{d,i}.$$

Figure 4.4 shows MERF of position as a function of α. The point at which MERF falls below 1 appears to be around $\alpha = 0.503$. As an approximation, consider if both delays were 0.1 seconds. Then, using equation (4.15) and taking only the real part of the poles (-0.22) as a, we can calculate an approximate analytical critical delay of $\tau_{r,crit} = 0.050275\ sec$, which implies $\alpha_{crit} = 0.50275$.

Example 2: Purely Imaginary Roots

Now, let us consider one of the special cases discussed earlier in the chapter. Suppose we remove the damping from system (4.15), so that we have

$$A = \begin{bmatrix} 0 & 1 \\ -0.6 & 0 \end{bmatrix}, \tag{4.16}$$

but leave all other parameters the same. The roots of this system are $\lambda = \pm 0.7746i$. From (4.18), we would expect the bound on acceptable delays to be

$$\alpha > 0.5.$$

Figure 4.5 shows MERF as a function of α for this new system. The first point at which MERF crosses below 1 is right around $\alpha = 0.5$, but at around $\alpha = 12$, it crosses back above 1.

Taking a test point at $\alpha = 15$ and solving equation (4.14) with initial condition $x_0 = \{1, 1\}$, we find that the left side of the equation is -3.1342, while the right side is -3.4198.

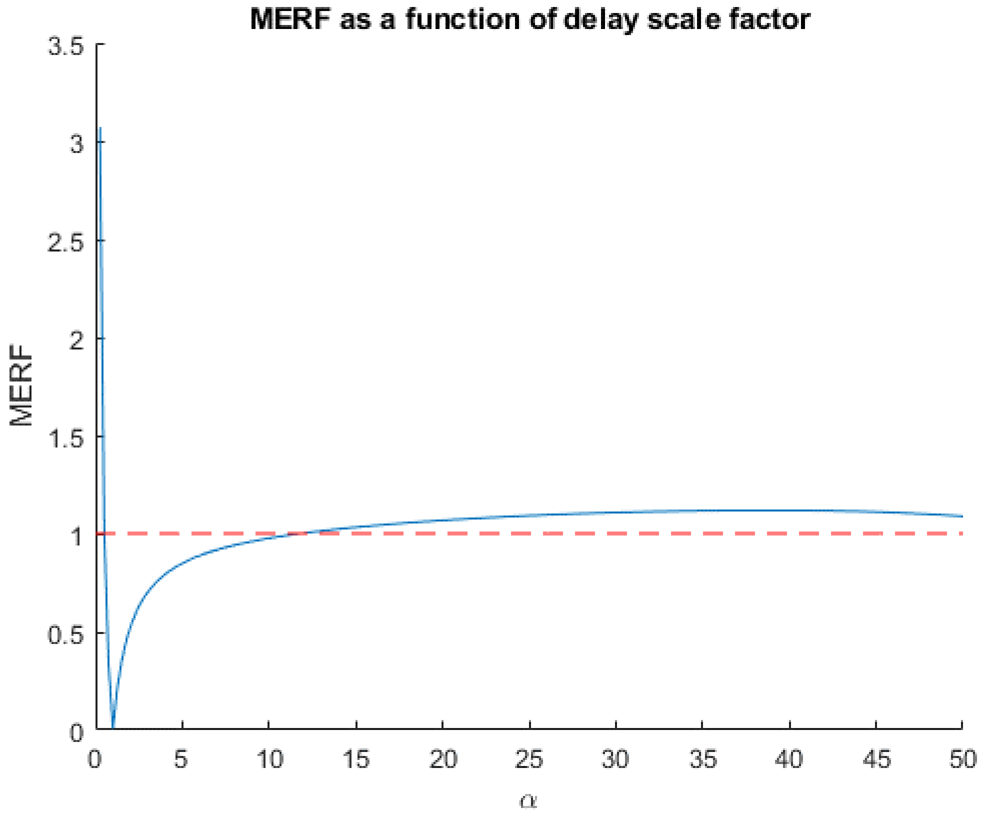

FIGURE 4.5. MERF for the undamped oscillator (4.16).

Since the left side is greater (less negative) than the right side, the equation predicts that the estimator fails at this particular point, which is borne out by the experimental result.

Conclusion

To summarize, for a general n-dimensional system with p different delays, a set of real delays $\{\tau_r\}$ can be checked against a set of expected delays $\{\tau_d\}$ to determine the performance of the DCE using (4.13) and (4.14). However, in the special case that the system is scalar and has only one measurement delay, the bounding delay $\tau_{r_{crit}}$ can be found from (4.15). Another special case is for second-order systems with a purely sinusoidal response, in which case the bound on acceptable delays is $\tau_r > \frac{\tau_d}{2}$.

CHAPTER 5

APPLICATION TO THE CONTROL OF A DYNAMIC SYSTEM USING ESTIMATED STATE FEEDBACK WITH MEASUREMENT DELAY

Chapter 3 developed the design and analysis process for the estimator in isolation, and chapter 4 developed delay bounds for the DCE based solely on the accuracy improvement gained by using an induced feedback delay. However, a common use for estimators is to enable full-state feedback for a control system, which comes with its own (often more stringent) requirements on the estimator accuracy.

Formulation

The control system we will consider is the following:

$$\dot{x}(t) = Ax(t) - BF\hat{x}(t)$$

$$\dot{\hat{x}}(t) = (A - BF)\hat{x}(t) + L\sum_{i=1}^{p} C_i\left[x\left(t - \tau_{r,i}\right) - \hat{x}\left(t - \tau_{d,i}\right)\right] \tag{5.1}$$

This is simply systems (1.1) and (4.1) combined with feedback control law $u(t) = -F\hat{x}(t)$. Because the measurement delay is assumed to be handled entirely by the DCE, the feedback gain F can be designed using established methods for un-delayed LTI systems, such as Ackermann's formula (for single input systems), robust pole placement (for multiple input systems) [12], or LQR and other optimal control methods.

Rewriting (5.1) into block matrix format, we have:

$$\begin{Bmatrix}\dot{x}(t)\\ \dot{\hat{x}}(t)\end{Bmatrix} = \begin{bmatrix}A & -BF\\ 0 & A - BF\end{bmatrix}\begin{Bmatrix}x(t)\\ \hat{x}(t)\end{Bmatrix} + \sum_{i=1}^{p}\left(\begin{bmatrix}0 & 0\\ LC_i & 0\end{bmatrix}\begin{Bmatrix}x\left(t - \tau_{r,i}\right)\\ \hat{x}\left(t - \tau_{r,i}\right)\end{Bmatrix} - \begin{bmatrix}0 & 0\\ 0 & LC_i\end{bmatrix}\begin{Bmatrix}x\left(t - \tau_{d,i}\right)\\ \hat{x}\left(t - \tau_{d,i}\right)\end{Bmatrix}\right) \tag{5.2}$$

This is a multiple-delay system with number of delays $= 2p$ and size $= 2n$. The rightmost poles of (5.2) can therefore be solved for using the Lambert W analysis method described in chapter 3.

This provides new criteria for success or failure of the DCE: when the poles of (5.2) are stable, the DCE is successful. Conversely, if the poles of (5.2) are unstable when the poles of the nominal controlled system $(A - BF)$ are stable, then the DCE has failed. If the poles of $(A - BF)$ were unstable to begin with, then we cannot make a judgement on the performance of the DCE.

Unfortunately, because the Lambert W method requires a numerical solution, it cannot be used to obtain an explicit function for the critical values of $\tau_{r,i}$ at which the system begins to fail. An exhaustive stability map in p-dimensional delay-space can be produced according to the Cluster Treatment of Characteristic Roots (CTCR) method outlined by Sipahi & Olgac [18]. This method is theoretically valid for systems of arbitrary size and arbitrary number of delays, and can reveal stability "islands" in delay-space that would otherwise be missed. While this method is elegant and very useful in an exhaustive robustness study, it requires the solution of a multinomial of order $2n$, which can get prohibitively complicated for large systems. Fortunately, it is often sufficient for a control designer to simply check the expected limits on $\{\tau_r\}$ (provided the range for $\{\tau_r\}$ is small) for stability by using the Lambert W method on (5.2) to find the rightmost poles.

Example Case: Observation and Control of a Double Inverted Pendulum on a Cart

As an illustrative example, consider the system in Figure 5.1 of two inverted pendulums balanced in sequence on top of a cart. The pendulums are free to swing around their hinges and the control input is an applied force to the cart.

System Dynamics

The dynamics of this system are derived in [31] using Lagrangian mechanics. See Appendix D for the full derivation. The resulting non-linear dynamics equations are:

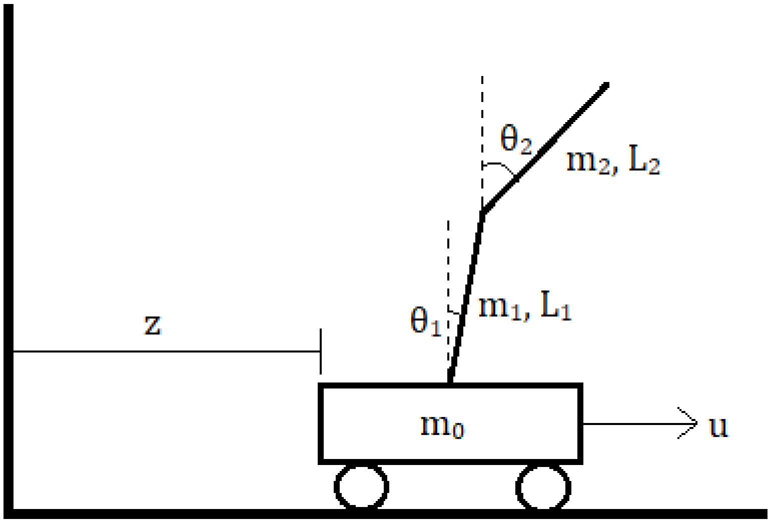

FIGURE 5.1. Schematic of the double inverted pendulum on a cart.

$$D(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta) = Hu \tag{5.3}$$

where

$$\theta = \begin{Bmatrix} z \\ \theta_1 \\ \theta_2 \end{Bmatrix}$$

$$D(\theta) = \begin{bmatrix} d_1 & d_2\cos\theta_1 & d_3\cos\theta_2 \\ d_2\cos\theta_1 & d_4 & d_5\cos(\theta_1-\theta_2) \\ d_3\cos\theta_2 & d_5\cos(\theta_1-\theta_2) & d_6 \end{bmatrix}$$

$$C(\theta,\dot{\theta}) = \begin{bmatrix} 0 & -d_2\sin(\theta_1)\,\dot{\theta}_1 & -d_3\sin(\theta_2)\,\dot{\theta}_2 \\ 0 & 0 & d_5\sin(\theta_1-\theta_2)\,\dot{\theta}_2 \\ 0 & -d_5\sin(\theta_1-\theta_2)\,\dot{\theta}_1 & 0 \end{bmatrix}$$

$$G(\theta) = \begin{Bmatrix} 0 \\ -f_1\sin\theta_1 \\ -f_2\sin\theta_2 \end{Bmatrix}$$

$$H = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}$$

The coefficients are given by

$$d_1 = m_0 + m_1 + m_2, \qquad d_2 = \left(\frac{1}{2}m_1 + m_2\right)L_1, \qquad d_3 = \frac{1}{2}m_2L_2$$

$$d_4 = \left(\frac{1}{3}m_1 + m_2\right)L_1^2, \qquad d_5 = \frac{1}{2}m_2L_1L_2, \qquad d_6 = \frac{1}{3}m_2L_2^2$$

$$f_1 = \left(\frac{1}{2}m_1 + m_2\right)L_1g, \qquad f_2 = \frac{1}{2}m_2L_2g$$

Linearizing (5.3) at $\theta_1 = \theta_2 = 0$ yields the following system of equations:

$$D_{lin}\ddot{\theta} + G_{lin}\theta = Hu$$

$$D_{lin} = \begin{bmatrix} d_1 & d_2 & d_3 \\ d_2 & d_4 & d_5 \\ d_3 & d_5 & d_6 \end{bmatrix}, \qquad G_{lin} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -f_1 & 0 \\ 0 & 0 & -f_2 \end{bmatrix}$$

Since D_{lin} is invertible, we can write

$$\ddot{\theta} = -D_{lin}^{-1}G_{lin}\theta + D_{lin}^{-1}Hu. \qquad (5.4)$$

Finally, writing (5.4) in linear state-space form yields the following state and input matrices:

$$x = \begin{Bmatrix} z \\ \dot{z} \\ \theta_1 \\ \dot{\theta}_1 \\ \theta_2 \\ \dot{\theta}_2 \end{Bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ T(1,1) & 0 & T(1,2) & 0 & T(1,3) & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ T(2,1) & 0 & T(2,2) & 0 & T(2,3) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ T(3,1) & 0 & T(3,2) & 0 & T(3,3) & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ U(1) \\ 0 \\ U(2) \\ 0 \\ U(3) \end{bmatrix}$$

where $T(i,j) = -D_{lin}^{-1}G_{lin}$ at the ith row and jth column, and $U(i) = D_{lin}^{-1}H$ at the ith row.

For this example, we will use the following values:

$$m_0 = 0.25\,kg, \qquad m_1 = 0.1\,kg, \qquad m_2 = 0.1\,kg$$

$$L_1 = 0.25\,m, \qquad L_2 = 0.2\,m, \qquad g = 9.81\,m/s^2$$

which gives the following linearized system:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -6.7915 & 0 & 0.7546 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 135.8308 & 0 & -54.3323 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -203.7462 & 0 & 169.7885 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 3.5897 \\ 0 \\ -18.4615 \\ 0 \\ 7.6923 \end{bmatrix}$$

Controller and Estimator Design

Implementation of the estimation and control scheme of (5.1) required the selection of two gain matrices: the feedback controller gain F and the estimator feedback gain L. The controller gain F was designed using the `place` command in MATLAB, which utilizes the pole placement algorithm in [12]. The estimator gain L was designed using the Lambert W method outlined in chapter 3.

Measurement delay modeling: To model the measurement delays, each state was assumed to be directly available to the DCE with a different delay. The cart position was assumed to be instantly available via an encoder on the wheels, while the cart velocity was assumed to be internally calculated by differentiating the position with a calculation delay τ_2. The velocities of the pendulum links were assumed to be the primary measurements via radar sensors, with delays $\{\tau_4, \tau_6\}$ for the states $\{\dot{\theta}_1, \dot{\theta}_2\}$. The pendulum link positions were assumed to be internally calculated based on the velocity measurements, with delays $\{\tau_3, \tau_5\}$ for states $\{\theta_1, \theta_2\}$ which were slightly higher than the corresponding velocity measurement delays. Since each measurement has a different delay, the corresponding C_i matrices are:

$$C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \ldots \; C_6 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

It is evident that $\sum C_i = I$, which has full rank, so the system is observable by the DCE.

Lacking any specifications for actual measurement devices, for this simulated experiment, the delays were arbitrarily picked as

$$\{\tau_d\} = \{0;\ 0.011;\ 0.022;\ 0.013;\ 0.028;\ 0.019\}.$$

This represents a computation delay of 0.011s for the cart velocity, radar measurement delays of 0.013s and 0.019s for the velocities of pendulum links 1 and 2, and an additional 0.009s computation delay for the pendulum positions (for a total of 0.021s and 0.028s for links 1 and 2). In a real-life application, however, the nominal delays would be approximated based on the response time of the specific measuring equipment being used.

An important note is that because $\tau_{d,1} = 0$, we cannot select $q = 1$ when using the multiple-delay Lambert W method.

Pole selection: The poles for the controller were selected somewhat arbitrarily to be $\lambda_c = [-2, -2.4, -8.5, -8.6, -9, -10]$. Since the rightmost controller pole was -2, the rightmost estimator pole was selected to be 5 times faster at $\lambda_e = -10$, while the rest of the estimator poles were left free to facilitate convergence of the design algorithm. The gain matrices designed for these pole placements were

$$F = [1.1850 \quad 1.6136 \quad -35.2523 \quad -0.5901 \quad 39.2853 \quad 3.0958]$$

$$L = \begin{bmatrix} 10 & 0.9648 & -0.0010 & -0.0001 & 0.0071 & 0.0005 \\ 0 & 10.1249 & -4.9902 & -0.0024 & 1.0876 & 0.0249 \\ 0 & 0.0025 & 9.3991 & 0.7521 & 6.1836 & 0.2826 \\ 0 & 0.0009 & 101.9034 & 11.0203 & -57.7001 & -0.4794 \\ 0 & 0.0005 & 0.0613 & -0.0130 & 8.4658 & 0.7793 \\ 0 & -0.0077 & -156.9518 & -0.2546 & 165.4453 & 12.0881 \end{bmatrix}$$

with the full eigenvalue spectrum of the principal branch S_0 of the estimator being

$$\lambda_{DCE} = [-10, -11.5047 \pm 0.0055i, -11.9983, -13.2826 \pm 0.5615i]$$

Simulation Setup

The simulation was performed using MATLAB and Simulink. A diagram of the simulation setup is shown in Figures 5.2 and 5.3. The full non-linear pendulum dynamics were

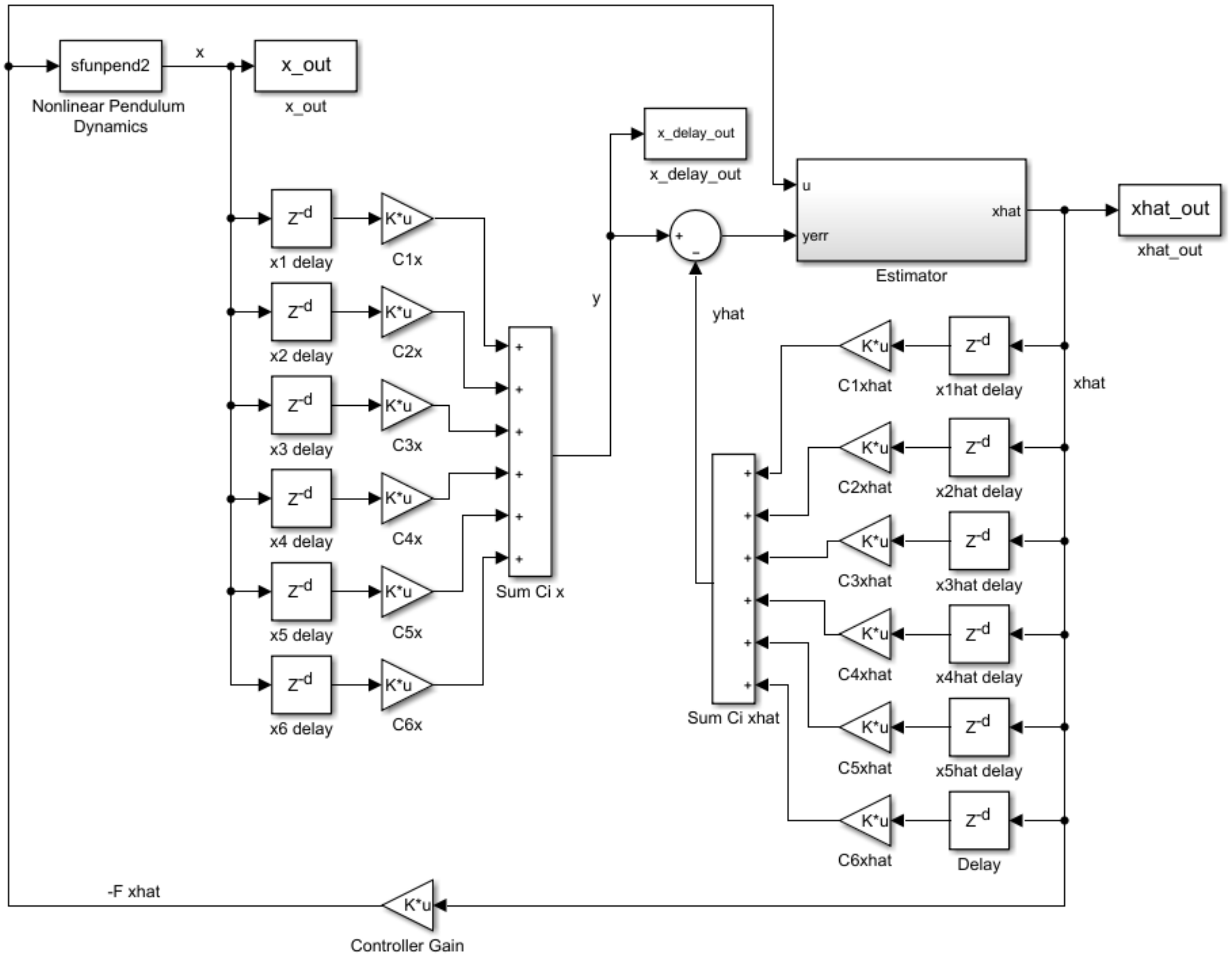

FIGURE 5.2. Main simulation block diagram in Simulink.

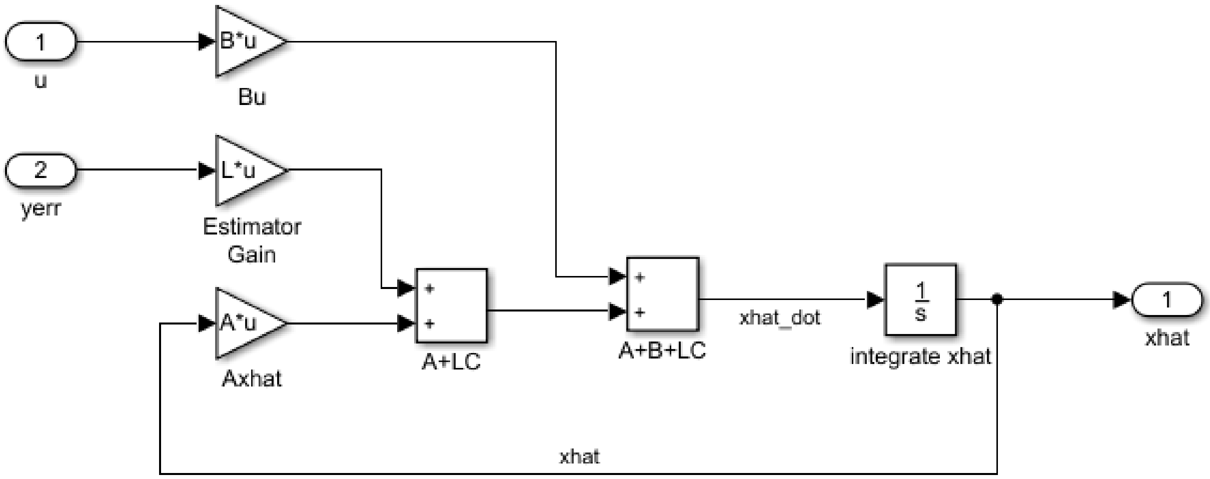

FIGURE 5.3. Internal block diagram of the estimator.

simulated, while the DCE only used the linearized dynamics. The initial conditions for the pendulum and estimator were

$$x_0 = \begin{pmatrix} 0 \\ 0.2 \\ 0.01 \\ 0 \\ 0.01 \\ 0 \end{pmatrix}, \quad \hat{x}_0 = \begin{pmatrix} 0.1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

The system was simulated for various real delays $\{\tau_r\}$, representing different errors in our original approximation of the delay.

Simulation Results

The simulation was tested at many different conditions, but only four cases for the real delays will be shown here as illustrative examples of the general trends:

Case 1: $\{\tau_r\} = \{\tau_d\} = \{0;\ 0.011;\ 0.022;\ 0.013;\ 0.028;\ 0.019\}$. This is the ideal case, and shows that given perfect knowledge of the delays, the system performs exactly as it was designed to.

Case 2: $\{\tau_r\} = \{0.001;\ 0.0111;\ 0.024;\ 0.0142;\ 0.0261;\ 0.0208\}$. This represents small, random errors in the approximation of the delays. It shows that a reasonable error in delay modeling produces a small depreciation in controller and estimator performance, with slightly longer settling time. The system does, however, remain stable. A Lambert W analysis of the principal branch of (5.2) reveals that the rightmost roots of the combined system have moved to $\lambda_{sys} = [-1.0748 \pm 1.7709i, -1.1854]$.

Case 3: $\{\tau_r\} = \{0.002;\ 0.017;\ 0.033;\ 0.020;\ 0.042;\ 0.029\}$. This represents a significant under-estimation of the actual delays present in the system. It shows that a large enough error in delay will, in fact, drive the system to instability. The rightmost roots for this case are $\lambda_{sys} = [0.1414 \pm 13.3835i]$.

Case 4: $\{\tau_r\} = \{0;\ 0;\ 0;\ 0;\ 0;\ 0\}$. This represents (the limit of) a system with delays much lower than expected. The controller performance is clearly much worse than Cases 1 and 2, with larger overshoot and slower settling time. For some systems and estimator designs, lower-than

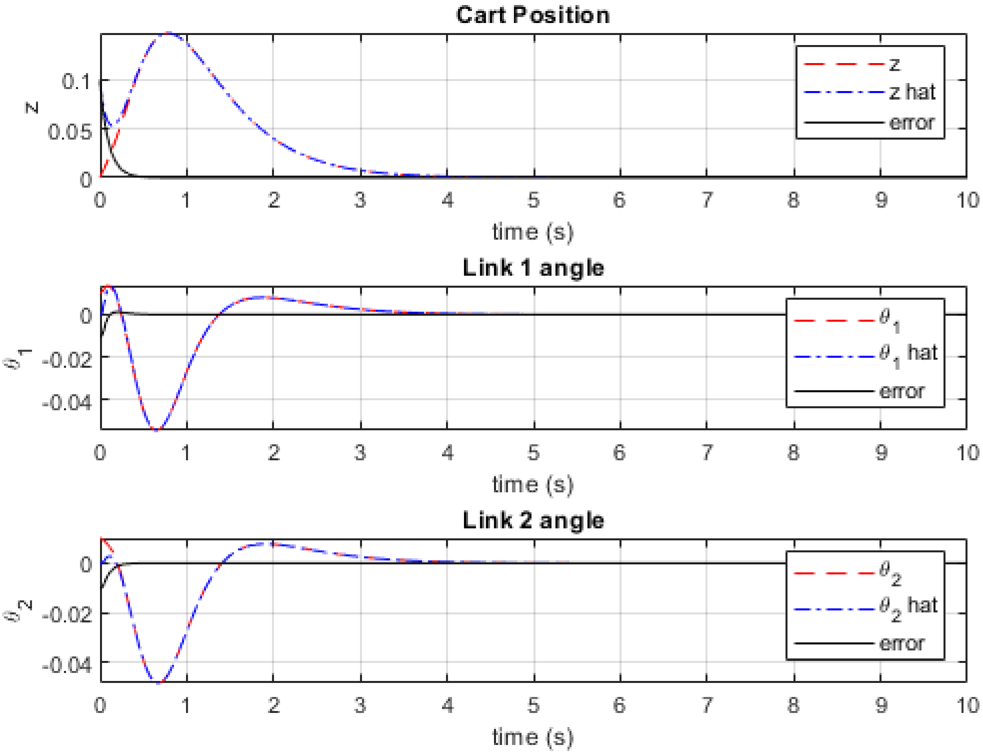

FIGURE 5.4. System response of Case 1 (ideal DCE).

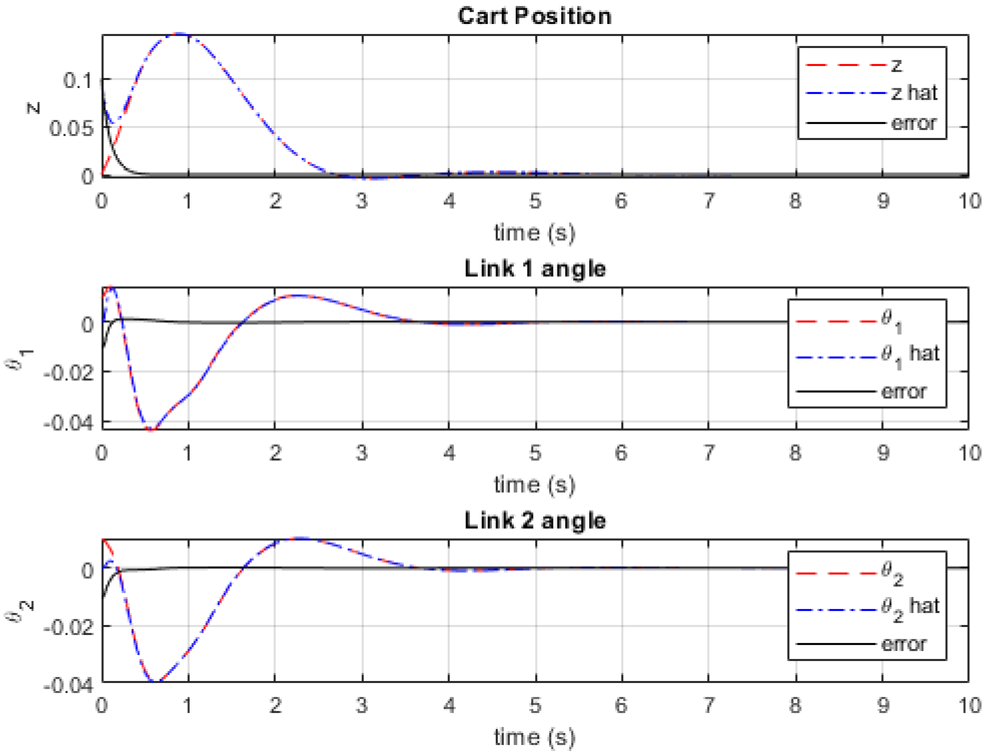

FIGURE 5.5. System response of Case 2 (small delay uncertainty).

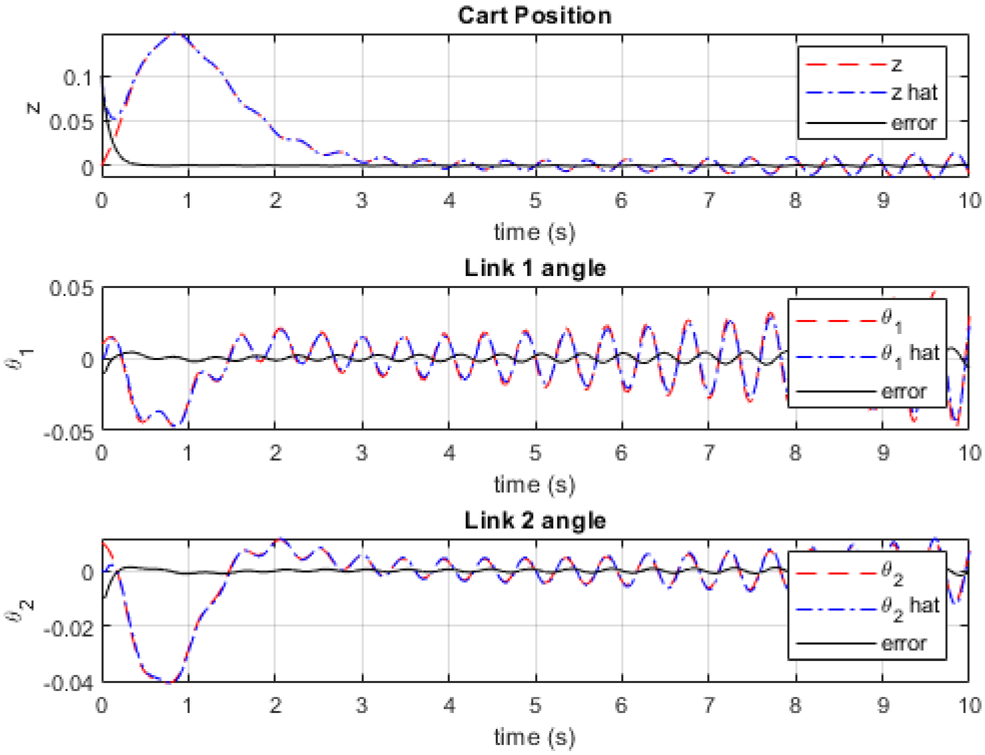

FIGURE 5.6. System response for Case 3 (large delay modeling errors).

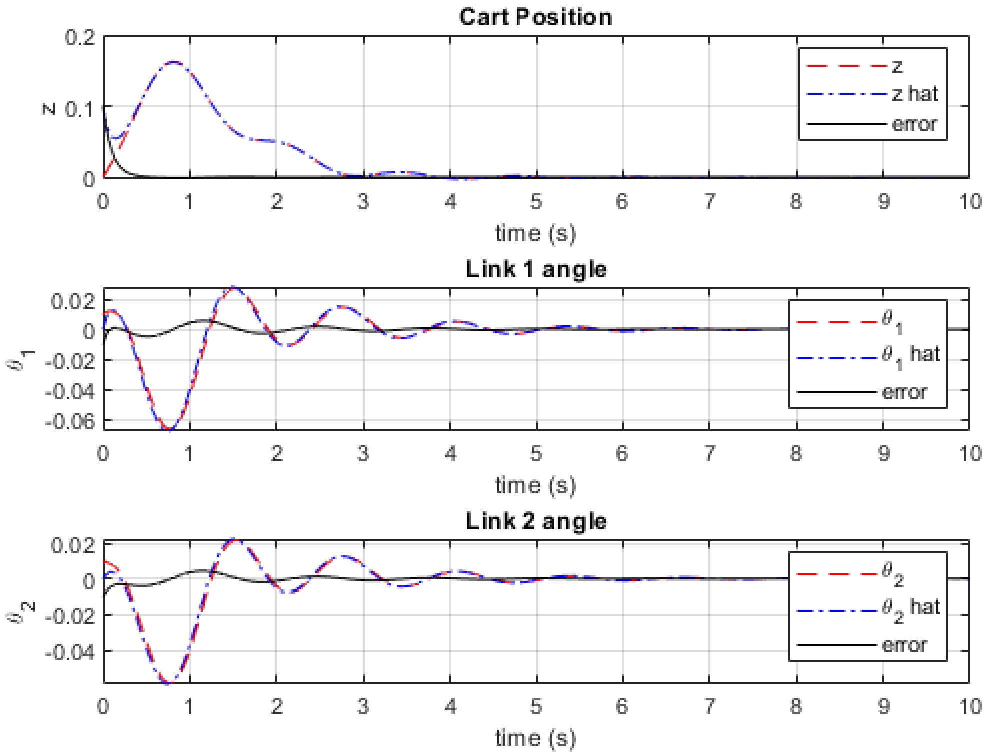

FIGURE 5.7. System response for Case 4 (no real delay).

-expected delays can actually drive the system to instability, but this particular example remains stable even at zero delay. The rightmost roots for this case are $\lambda_{sys} = [-0.6591 \pm 4.82i]$.

The cart position and pendulum angles are plotted in Figure 5.3-5.7 for each of the four cases. For comparison, Figure 5.8 shows the system response if a Luenberger observer with the same designed poles had been used instead of the DCE with the same real delays $\{\tau_r\}$ as Case 1. Clearly, the delays are outside the robust stability radius of the Luenberger observer, so the DCE is needed in this case.

Conclusion

As shown through the example problem, the DCE can be used as an observer to control a dynamic system subject to multiple measurement delays, including some un-delayed information. The robustness of this design with respect to uncertainty in the delays is in general

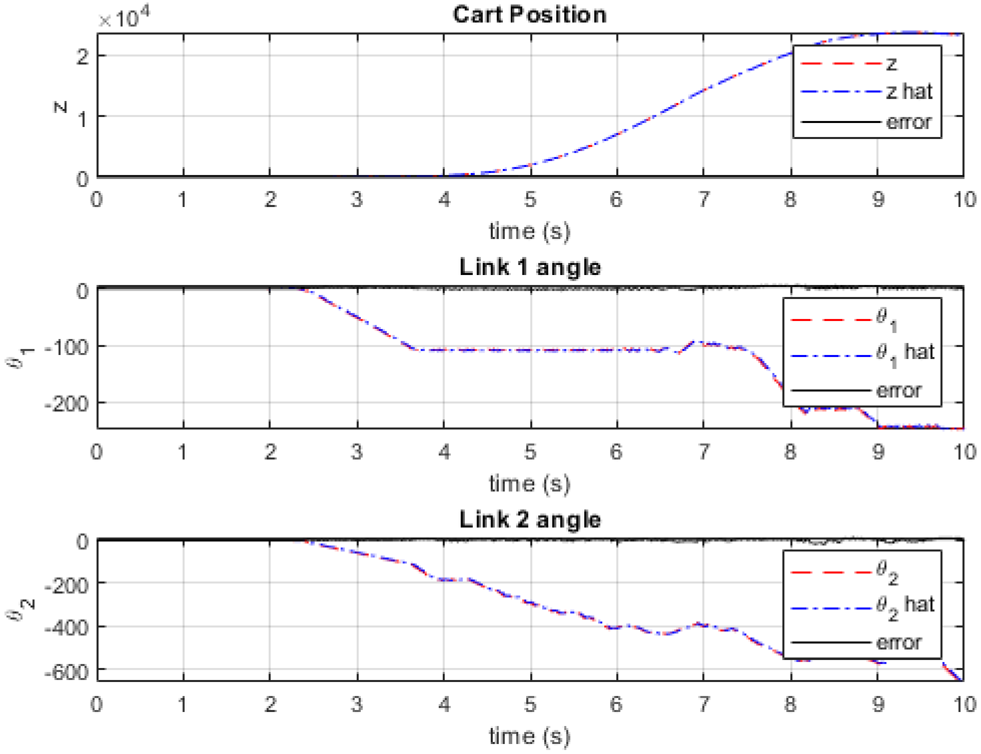

FIGURE 5.8. System response with a Luenberger observer instead of the DCE.

dependent upon the system in question, but is no better or worse than the sensitivity of the classic Luenberger observer with respect to delays. The major advantage provided by the DCE, then, is that a normally unstable delayed observer-controller scheme is stabilizable if the delays can be approximated well enough, and DCE-controller performance can be improved as the delays are known more precisely. The only drawback is that all states must be measurable (i.e. $rank(\sum C_i) = n$) for the DCE to be designable.

CHAPTER 6

CONCLUSION AND FUTURE WORK

The Delay-Compensating Estimator (DCE) was developed as a method of filtering out the effects of multiple different delays in measurement data to recover the un-delayed states of the system being measured. This method can be used by itself to improve the accuracy measurement data, or (as was shown in Chapter 5) it can be used in tandem with a standard state feedback controller to stabilize a dynamic system with multiple measurement delays which would otherwise have been unstable with a traditional observer-controller pair. The major drawback of the DCE is that it requires the sum of all the measurement matrices C_i to have rank n, which means it cannot be used with any system that has un-measurable internal states.

In pursuit of a way to design the feedback gain L of the DCE, an extension of the Lambert W method to include systems with multiple delays was developed in Chapter 3. This extended method was able to find the rightmost roots of multiple time-delay systems, and enabled the design of the DCE feedback gain. However, the extended method still lacks a reliable initial guess to analyze the non-principal branches of the Lambert W function, and cannot yet be used to calculate the full root spectrum of a multiple-delay system.

Future Work

From a theoretical standpoint, the biggest gap in the methods presented here is the current inability to use the extended Lambert W method to analyze non-principal branches of multiple-delay systems. If appropriate initial guesses are found, the extended Lambert W method could become a unified framework for analysis of all LTI systems with an arbitrary number of time delays.

In terms of practical applications, the major downside of the DCE is that it requires all states to be measurable (either directly or indirectly). A partial solution to this problem was envisioned, but not explored in this thesis: use a Luenberger observer to reconstruct the full states from partial measurements, and then use the DCE to filter out the delays from the observed states. This presents a potential avenue for future research.

Other potential future work of interest includes an extension of the DCE to estimate the real measurement delays $\{\tau_r\}$ in real time, and update the induced delays $\{\tau_d\}$ accordingly to improve performance. The DCE could also be re-formulated for stochastic systems to augment a Kalman filter and reduce measurement uncertainty due to delays.

It is hoped that the concept outlined in this thesis will be useful to future studies of both time-delay systems and estimation theory.

APPENDICES

APPENDIX A

INFINITE ROOT SPECTRUM OF A TIME DELAY SYSTEM

APPENDIX A

INFINITE ROOT SPECTRUM OF A TIME DELAY SYSTEM

A linear time-invariant (LTI) retarded time delay system can be written in state-space form as

$$\dot{x}(t) = Ax(t) + A_d x(t - \tau)$$

The roots of this system can be found by taking the Laplace transform,

$$\mathcal{L}\{\dot{x}(t) = Ax(t) + A_d x(t - \tau)\}$$

$$sX(s) - x(0) = AX(s) + A_d X(s) e^{-s\tau},$$

and then solving for $X(s)$, yielding

$$X(s) = (sI - A - A_d e^{-s\tau})^{-1} x(0).$$

The characteristic equation for this system is

$$sI - A - A_d e^{-s\tau} = 0.$$

Solving the characteristic equation for s yields the roots. However, because of the exponential term, there are an infinite number of complex numbers s which will solve the characteristic equation. The exponential term is a direct result of taking the Laplace transform of the time-delayed term, so we can conclude that any system with time delays will have an infinite number of roots.

APPENDIX B

EQUIVALENCE OF MEASUREMENT DELAYS AND CONTROLLER INPUT DELAYS IN IDEAL FEEDBACK CONTROL SYSTEMS

APPENDIX B

EQUIVALENCE OF MEASUREMENT DELAYS AND CONTROLLER INPUT DELAYS IN IDEAL FEEDBACK CONTROL SYSTEMS

Consider the LTI system with measurement delay τ_m and control input delay τ_c with output feedback control:

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

$$y(t) = Cx(t - \tau_m)$$

$$u(t) = Fy(t)$$

The delayed control input $u(t - \tau_c)$ can be rewritten as

$$u(t - \tau_c) = FCx\big((t - \tau_c) - \tau_m\big) = FCx\big(t - (\tau_c + \tau_m)\big).$$

The ideal system dynamics are therefore

$$\dot{x}(t) = Ax(t) + BFCx\big(t - (\tau_c + \tau_m)\big).$$

It is evident from this expression that τ_c and τ_m are indistinguishable in terms of the closed-loop system dynamics, hence why they can generally be lumped together for the purpose of control system design.

APPENDIX C

SOLUTION OF INTEGRAL (4.12)

APPENDIX C

SOLUTION OF INTEGRAL (4.12)

The solution to the integral

$$\Psi_{ij} = \int_0^\infty (e^{At})' C_i' C_j e^{At} dt \tag{4.12}$$

is presented via integration by parts. Let

$$u = e^{A't} C_i'$$

$$dv = C_j e^{At} dt.$$

Then,

$$du = A' e^{A't} C_i' dt$$

$$v = C_j e^{At} A^{-1}.$$

By the definition of integration by parts,

$$\Psi_{ij} = [uv]_0^\infty - \int_0^\infty du * v = \left[e^{A't} C_i' C_j e^{At} A^{-1}\right]_0^\infty - \int_0^\infty A' e^{A't} C_i' C_j e^{At} A^{-1} dt.$$

If all eigenvalues of A have negative real parts, then $\lim_{t\to\infty} e^{At} = 0 = \lim_{t\to\infty} e^{A't}$. We can then write

$$\Psi_{ij} = -C_i' C_j A^{-1} - A' \Psi_{ij} A^{-1}.$$

Post-multiplying by A and rearranging,

$$A' \Psi_{ij} + \Psi_{ij} A + C_i' C_j = 0.$$

APPENDIX D

DERIVATION OF THE DYNAMICS OF A DOUBLE INVERTED PENDULUM ON A CART

APPENDIX D

DERIVATION OF THE DYNAMICS OF A DOUBLE INVERTED PENDULUM ON A CART

Following the derivation in [31], the Lagrangian is defined as the kinetic energy of the system minus the potential energy:

$$L = KE - PE.$$

For the system in Figure 5.1, the kinetic energies of each component are:

$$KE_{cart} = \frac{1}{2} m_0 \dot{z}^2$$

$$KE_{L1} = \frac{1}{2} m_1 \left[\left(\dot{z} + \frac{L_1}{2} \dot{\theta}_1 \cos\theta_1 \right)^2 + \left(\frac{L_1}{2} \dot{\theta}_1 \sin\theta_1 \right)^2 \right] + \frac{1}{2} I_1 \dot{\theta}_1^2$$

$$KE_{L2} = \frac{1}{2} m_2 \left[\left(\dot{z} + L_1 \dot{\theta}_1 \cos\theta_1 + \frac{L_2}{2} \dot{\theta}_2 \cos\theta_2 \right)^2 + \left(L_1 \dot{\theta}_1 \sin\theta_1 + \frac{L_2}{2} \dot{\theta}_2 \sin\theta_2 \right)^2 \right] + \frac{1}{2} I_2 \dot{\theta}_2^2$$

and the potential energies are:

$$PE_{cart} = 0$$

$$PE_{L1} = m_1 g \frac{L_1}{2} \cos\theta_1$$

$$PE_{L2} = m_2 g \left(L_1 \cos\theta_1 + \frac{L_2}{2} \cos\theta_2 \right)$$

The dynamics equations which describe the system are then

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = \begin{Bmatrix} u \\ 0 \\ 0 \end{Bmatrix}$$

where $q = \{z, \theta_1, \theta_2\}^T$ are the generalized coordinates of the system. The partial derivatives are given by:

$$\frac{\partial L}{\partial z} = 0$$

$$\frac{\partial L}{\partial \theta_1} = -\left(\frac{1}{2}m_1 + m_2\right)L_1\dot{z}\dot{\theta}_1 \sin\theta_1 - m_2 L_1 \frac{L_2}{2}\dot{\theta}_1\dot{\theta}_2 \sin(\theta_1 - \theta_2) + \left(\frac{1}{2}m_1 + m_2\right) g L_1 \sin\theta_1$$

$$\frac{\partial L}{\partial \theta_2} = -m_2 \frac{L_2}{2}\dot{z}\dot{\theta}_2 \sin\theta_2 + m_2 L_1 \frac{L_2}{2}\dot{\theta}_1\dot{\theta}_2 \sin(\theta_1 - \theta_2) + m_2 g \frac{L_2}{2}\sin\theta_2$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{z}}\right) = (m_0 + m_1 + m_2)\ddot{z} + \left(\frac{1}{2}m_1 + m_2\right)L_1\left(\ddot{\theta}_1 \cos\theta_1 - \dot{\theta}_1^2 \sin\theta_1\right)$$

$$+ m_2 \frac{L_2}{2}\left(\ddot{\theta}_2 \cos\theta_2 - \dot{\theta}_2^2 \sin\theta_2\right)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) = \left(\frac{1}{2}m_1 + m_2\right)L_1\left(\ddot{z}\cos\theta_1 - \dot{z}\dot{\theta}_1 \sin\theta_1\right) + \left(m_1 \frac{L_1^2}{4} + m_2 L_1 + I_1\right)\ddot{\theta}_1$$

$$+ m_2 L_1 \frac{L_2}{2}\left(\ddot{\theta}_2 \cos(\theta_1 - \theta_2) - \dot{\theta}_2\left(\dot{\theta}_1 - \dot{\theta}_2\right)\sin(\theta_1 - \theta_2)\right)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) = m_2 \frac{L_2}{2}\left(\ddot{z}\cos\theta_2 - \dot{z}\dot{\theta}_2 \sin\theta_2\right) + \left(m_2 \frac{L_2^2}{4} + I_2\right)\ddot{\theta}_2$$

$$+ m_2 L_1 \frac{L_2}{2}\left(\ddot{\theta}_1 \cos(\theta_1 - \theta_2) - \dot{\theta}_1\left(\dot{\theta}_1 - \dot{\theta}_2\right)\sin(\theta_1 - \theta_2)\right)$$

Here, $I_n = \frac{1}{12}m_n L_n^2$ is the moment of inertia of the pendulum arms. Plugging in the partial derivatives yields a system of three non-linear equations, each second-order with respect to all three coordinates. This system can be written in compact form as

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = Hu$$

$$D(q) = \begin{bmatrix} d_1 & d_2\cos\theta_1 & d_3\cos\theta_2 \\ d_2\cos\theta_1 & d_4 & d_5\cos(\theta_1 - \theta_2) \\ d_3\cos\theta_2 & d_5\cos(\theta_1 - \theta_2) & d_6 \end{bmatrix}$$

$$C(q,\dot{q}) = \begin{bmatrix} 0 & -d_2\sin(\theta_1)\,\dot{\theta}_1 & -d_3\sin(\theta_2)\,\dot{\theta}_2 \\ 0 & 0 & d_5\sin(\theta_1 - \theta_2)\,\dot{\theta}_2 \\ 0 & -d_5\sin(\theta_1 - \theta_2)\,\dot{\theta}_1 & 0 \end{bmatrix}$$

$$G(q) = \begin{Bmatrix} 0 \\ -f_1\sin\theta_1 \\ -f_2\sin\theta_2 \end{Bmatrix}$$

$$H = \begin{Bmatrix} 1 \\ 0 \\ 0 \end{Bmatrix}$$

with coefficients given by

$$d_1 = m_0 + m_1 + m_2, \qquad d_2 = \left(\frac{1}{2}m_1 + m_2\right)L_1, \qquad d_3 = \frac{1}{2}m_2 L_2$$

$$d_4 = \left(\frac{1}{3}m_1 + m_2\right)L_1^2, \qquad d_5 = \frac{1}{2}m_2 L_1 L_2, \qquad d_6 = \frac{1}{3}m_2 L_2^2$$

$$f_1 = \left(\frac{1}{2}m_1 + m_2\right)L_1 g, \qquad f_2 = \frac{1}{2}m_2 L_2 g$$

To linearize these equations about the equilibrium point $\theta_1 = \theta_2 = 0$, we make the following substitutions:

$$\cos\theta_1 = \cos\theta_2 = \cos(\theta_1 - \theta_2) = 1$$

$$\sin\theta_1 = \theta_1, \qquad \sin\theta_2 = \theta_2, \qquad \sin(\theta_1 - \theta_2) = \theta_1 - \theta_2$$

$$\theta\dot{\theta} = 0$$

It is easy to see that with these substitutions, $D(q)$ reduces to

$$D_{lin} = \begin{bmatrix} d_1 & d_2 & d_3 \\ d_2 & d_4 & d_5 \\ d_3 & d_5 & d_6 \end{bmatrix}$$

and $C(q, \dot{q})$ reduces to zero. Similarly, $G(q)$ reduces to

$$G(q) \to G_{lin}q = \begin{Bmatrix} 0 \\ -f_1\theta_1 \\ -f_2\theta_2 \end{Bmatrix}.$$

REFERENCES

REFERENCES

[1] Mahmoud, M. S., 2017, "Recent Progress in Stability and Stabilization of Systems with Time-Delays," Math. Problems in Eng., pp. 1-25.

[2] Bellman, R., and Cooke, K. L., 1963, "Differential Difference Equations," R-374-PR.

[3] Liu, K., Ma, Q., Liu, H., Cao, Z., and Liu, Y., 2013, "End-to-end Delay Measurement in Wireless Sensor Networks without Synchronization," Proceedings of the IEEE 10th Int'l Conf. on Mobile Ad-Hoc and Sensor Sys., Hangzhou, China, October 14-16, 2013, pp. 583-591.

[4] Yi, S., Nelson, P. W., and Ulsoy, A. G, 2010, "Eigenvalue Assignment via the Lambert W Function for Control of Time-Delay Systems," J. Vibration Control, 16(7-8), pp. 961-982.

[5] Hua, C., Guan, X., and Shi, P., 2005, "Robust Adaptive Control for Uncertain Time-Delay Systems," Int'l J. Adaptive Control and Signal Processing, 19, pp. 531-545.

[6] Michiels, W., Vyhlidal, T., and Zitek, P., 2010, "Control Design for Time-Delay Systems Based on Quasi-Direct Pole Placement," J. Process Control, 20, pp. 337-343.

[7] Zhong, Q., 2005, Robust Control of Time-Delay Systems, Springer, Berlin.

[8] Orguner, U., and Gustafsson, F., 2011, "Target Tracking with Particle Filters Under Signal Propagation Delays," IEEE Transactions on Signal Processing, 59(6), pp. 2485-2495.

[9] Jiang, H., van der Veek, B., Kirk, D., and Gutierrez, H., 2013, "Real-Time Estimation of Time-Varying Bending Modes Using Fiber Bragg Grating Sensor Arrays," AIAA Journal, 51(1), pp. 178-185.

[10] Rezaei, R., Ghabrial, F., Besnard, E., Shankar, P., Castro, J., Labonte, L., Razfar, M., and Abedi, A., 2013, "Determination of Elastic Mode Characteristics Using Wirelessly Networked Sensors for Nanosat Launch Vehicle Control," Proceedings of the IEEE Int'l Conf. on Wireless for Space and Extreme Environments, Baltimore, MD, November 7-9, 2013, pp. 1-2.

[11] Luenberger, D. G., 1979, Introduction to Dynamic Systems: Theory, Models, and Applications, John Wiley & Sons, New York.

[12] Kautsky, J., Nichols, N. K., and Van Dooren, P., 1985, "Robust Pole Assignment in Linear State Feedback," Int'l J. Control, 41(5), pp. 1129-1155.

[13] Smith, O. J. M., 1957, "Closer Control of Loops with Dead Time," Chem. Eng. Progress, 53(5), pp. 217-219.

[14] Sourdille, P. and O'Dwyer, A., 2003, "An Outline and Further Development of Smith Predictor Based Methods for Compensation of Processes with Time Delay," Proceedings of the Irish Signals and Systems Conf., Limerick, Ireland, July 1, 2003, pp. 338-343.

[15] Adam, E. J., Latchman, H. A., and Crisalle, O. D., 2000, "Robustness of the Smith Predictor with Respect to Uncertainty in the Time-Delay Parameter," Proceedings of the American Control Conf., Chicago, IL, June 2000, pp. 1452-1457.

[16] Sename, O., 1997, "Unknown Input Robust Observers for Time-Delay Systems," Proceedings of the 36th Conf. on Decision and Control, San Diego, CA, December 1997, pp. 1629-1630.

[17] Olgac, N., and Sipahi, R., 2002, "An Exact Method for the Stability Analysis of Time-Delayed Linear Time-Invariant (LTI) Systems," IEEE Trans. Automatic Control, 47(5), pp. 793-797.

[18] Sipahi, R., and Olgac, N., 2006, "A Unique Methodology for the Stability and Robustness of Multiple Time Delay Systems," Systems and Control Let., 55, pp. 819-825.

[19] Asl, F. M., and Ulsoy, A. G., 2003, "Analysis of a System of Linear Delay Differential Equations," J. Dyn. Syst-T. ASME, 125(2), pp. 215-223.

[20] Yi, S., and Ulsoy, A. G., 2006, "Solution of a System of Linear Delay Differential Equations Using the Matrix Lambert Function," Proceedings of the American Control Conf., Minneapolis, MN, June 14-16, 2006, pp. 2433-2438.

[21] Corless, R. M., Gonnet, G. H., Hare, D.E.G, Jeffrey, D.J., and Knuth, D. E., 1996, "On the Lambert W Function," Adv. in Computational Mathematics, 5(1), pp. 329-359.

[22] Mathworks, Symbolic Toolbox: User's Guide (R2018b), 2018.

[23] Fasi, M., Higham, N. J., and Iannazzo, B., 2015, "An Algorithm for the Matrix Lambert W Function," SIAM J. Matrix Analysis Applications, 36(2), pp. 669-685.

[24] Yi, S., 2009, "Time Delay Systems: Analysis and Control Using the Lambert W Function," Ph.D. dissertation, University of Michigan, Ann Arbor, MI.

[25] Shinozaki, H., and Mori, T., 2006, "Robust Stability Analysis of Linear Time-Delay Systems by Lambert W Function: Some Extreme Point Results," Automatica, 42, pp. 1781-1799.

[26] Mathworks, Optimization Toolbox: User's Guide (R2018b), 2018.

[27] Yi, S., Nelson, P. W., and Ulsoy, A. G., 2008, "Controllability and Observability of Systems of Linear Delay Differential Equations via the Matrix Lambert W Function," IEEE Trans. on Automatic Control, 53(3), pp. 854-860.

[28] Cepeda-Gomez, R., and Michiels, W., 2015, "Some Special Cases in the Stability Analysis of Multi-Dimensional Time Delay Systems Using the Matrix Lambert W Function," Automatica, 53, pp. 339-345.

[29] Ivanoviene, I., and Rimas, J., 2015, "Complement to Method of Analysis of Time Delay Systems via the Lambert W Function," Automatica, 54, pp. 25-28.

[30] Li, X. R., and Zhao, Z., 2001, "Measures of Performance for Evaluation of Estimators and Filters," SPIE Conf. on Signal and Data Processing of Small Targets, San Diego, CA, November 26, 2001, pp. 1-12.

[31] Bogdanov, A., 2004, "Optimal Control of a Double Inverted Pendulum on a Cart," Technical Report No. CSE-04-006.