TIME-VARYING FEEDBACK SYSTEMS DESIGN VIA DIOPHANTINE

EQUATION ORDER REDUCTION

by

SHR-HUA WU

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2007

ACKNOWLEDGEMENTS

ABSTRACT

TIME-VARYING FEEDBACK SYSTEMS DESIGN VIA DIOPHANTINE

EQUATION ORDER REDUCTION

Publication No. _____

Shr-Hua Wu, PhD.

The University of Texas at Arlington, 2007

Supervising Professor: Kai-Shing Yeung

Diophantine equation plays an important role in the design and synthesis of feedback compensators. Many methods have been developed to solve the Diophantine equation. This dissertation develops a new systematic approach of solving a linear time-varying Diophantine equation. This approach is based upon successively reducing the order of the Diophantine equation by Euclidean algorithm. Euclidean algorithm for solving for both time-invariant and time-varying Diophantine equations for directly determining both the quotient and the remainder associated with the division of one polynomial by another is presented. The coprimeness (right or left) of two Polynomial Differential Operators is needed to guarantee, in general, the existence of solutions of the respective Diophantine equation. The illustrative examples are given.

This dissertation also develops a procedure of setting up canonical forms for linear time-varying, single-input single-output systems. The starting point is a differential equation description of the system. Two canonical forms are considered: observability and observer forms. Initial condition conversions between the canonical forms and the differential equation description are also derived.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Review of Diophantine Equation

Diophantine equation plays an important role in the design and synthesis of compensators [1][2]. Solving Diophantine equation is shown to be equivalent to design a full or reduced-order observer of a linear functional of the state [3].The Diophantine equation approach is a transfer function-based control theory in which the transfer functions are viewed and handled as algebraic objects. Solving a system of linear Diophantine equations is a classical mathematical problem as well. Computing integer solutions to systems of linear Diophantine equations is a classical mathematical problem with many interesting applications in number theory [4], group theory [5] and combinatorics [6].

Many methods have been developed to solve the Diophantine equation. An indirect method of its solution is the well-known combined observer-controller design [1]. Another method is the method developed by C.T. Chen [7]. Some of the methods used involve the state-space approach [8] and [9]. Others utilize the coefficient matching approach [7], [10] and [11]. [12] generalizes a classical Knuth-Schonhage algorithm computing two Greatest Common Divisors (GCD) of two polynomials for solving two arbitrary linear Diophantine systems over polynomials in time, quasi-linear

in the maximal degree for the weighted curve-fitting problem. One popular problem encountered in the analysis and design of control systems is the solutions of the Diophantine equations in which the existence of solutions of the Diophantine equations requires the coprimeness of associated pairs of polynomial matrices [7]. The recursive algorithm is commonly applied to solve Polynomial Diophantine Equation. Both [13] and [14] propose a recursive algorithm for coprime fractions and Diophantine equation. The above-mentioned works are all focused on linear time-invariant systems.

Step forward in extending the Diophantine equation approach is the case of time-varying linear systems. The field of coefficients is replaced by a differential polynomial over an interval of time [15][16]. The time-varying coefficients are no longer commutative, for multiplication and differentiation do not commute. Solving the Sylvester equation is the most commonly used to solve Diophantine equation with time-varying coefficient matrices. Sylvester equations have been always the significant tool in the development of numerical algorithms for various feedback design techniques in control theory [17][18]19][20][21]. But Sylvester matrix has its limitations in MIMO system. These limitations can be overcome with Euclidean algorithm.

<u>1.2 Review of Euclidean Algorithm</u>

In this dissertation, Euclidean algorithm is presented for solving for both time-invariant and time-varying Diophantine equations for directly determining both the quotient and the remainder associated with the division of one polynomial by another. The coprimeness (right or left) of two Polynomial Differential Operators is needed to guarantee, in general, the existence of solutions of the respective Diophantine

2

equation[22]. The Euclidean algorithm has been used to solve Sylvester matrices [23] and Diophantine equations [24] in time-invariant. These methods can in principle often also lead to a GCD. GCD of polynomial matrices plays an important part in the theory and application of general differential systems as studied extensively by [25][26], and others. They are useful in obtaining irreducible matrix fraction descriptions (and hence minimal state-space realizations) of transfer-function matrices, studying decoupling zeros and uncontrollable and unobservable modes of given systems, and obtaining the pole-zero structure of given multivariable systems [27].

1.3 Review of Canonical Form

Canonical forms of state variable equations are of importance in the analysis and synthesis of dynamical systems. To simulate a continuous time system, there is a fundamental weakness to use differentiator. In practice, all signals are corrupted by noise. When such signal is differentiated, the derivative of the usually rapidly varying noise will shout down the derivative of the signal. For system described by a general differential equation the integrating device can be used as the basic building block for simulation[28]. Linear time-invariant canonical forms have been extensively researched and described in several papers and texts (see [1][7][29] and references therein). The controllability, observability, controller and observer canonical forms are frequently used. Nevertheless, there are only a few investigations related to time-varying canonical forms: Zeitz [30] derived observability form of non-linear time-varying systems from state variable equations, Krener [31] constructed observer forms for those nonlinear

3

systems which can be transformed into a linear system, and Schaft [32] represented a nonlinear state space system as a set of higher-order differential equations.

The general formulae relating the initial conditions of a scalar linear time-invariant differential equation to those of its state variable representation have been presented many years ago [33]. In this dissertation, a derivation of observer and observability canonical forms from a differential equation description is developed for linear time-varying systems. Initial condition conversions between the canonical forms and the differential equation description are given. With the help of the derived canonical forms, simulations of systems given by linear time-varying differential equations are greatly facilitated. This can be done, For example, using MATLAB software to implement integrators, summers and gains.

1.4 Outline

This dissertation demonstrates a new systematic approach to tackle both the time-invariant and time-varying Diophantine equations using an order reduction method. This method is based on the Euclidean algorithm. The time-invariant case is presented in chapter 2 and the time-varying case is discussed in chapter 3, we shall briefly review the pole-placement approaches and the Euclidean algorithm for both time-invariant and time-varying cases in both chapter 2 and 3, and how they require the solution of the Diophantine equation. In chapter 3 section 2, the division algorithm in the time-varying case is illustrated. Then, we shall introduce the Diophantine equation order reduction process, coupled with the use of the Euclidean algorithm. Examples will be given to illustrate the order reduction method for both time-invariant and time-

varying cases. Chapter 4 illustrates the basic ideas and procedures of setting up LTV canonical forms in both observability and observer. Formulas for the associated initial condition conversion are given as well. Chapter 5 shows case study utilizing Diophantine equation order reduction, LTV canonical forms and comparisons of existing input-state and input-output methods. The contribution of this dissertation and suggestions of future work are discussed in chapter 6.

CHAPTER 2

DIOPHANTINE EQUATION ORDER REDUCTION USING EUCLIDEAN
ALGORITHM

2.1 Pole-Placement Feedback Design

The pole-placement in a closed-loop structure using output feedback can be done by means of solving the Diophantine equation. The poles of the overall transfer function are assigned in order to meet some given performance requirements. Consider a unity feedback control systems as shown in Figure 2.1. This scheme is considered to be the simplest unity feedback configuration. Given a plant with a rational transfer function $H(s)$ of order k_0 with input u and output y and a reference signal r, the objective is to design a proper controller $C(s)$ of order γ_0 so that the closed-loop poles will reside at specified locations.
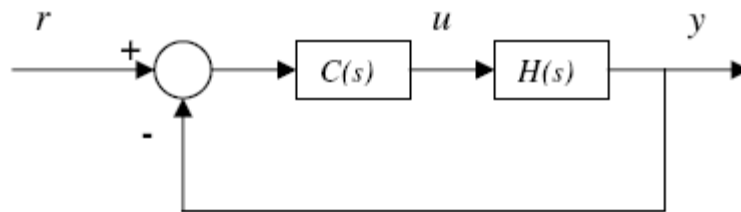
Figure 2.1. Unity feedback control scheme

Let the plant be given as a rational transfer function

$$H(s) = \frac{b(s)}{a(s)}, \qquad \deg[a(s)] > \deg[b(s)] \tag{2.1.1}$$

To design a controller $C(s)$ with a proper transfer function, let

$$C(s) = \frac{n(s)}{d(s)}, \qquad \deg[d(s)] \geq \deg[n(s)] \tag{2.1.2}$$

Then the overall transfer function from r to y is

$$\begin{aligned} T(s) &= \frac{C(s)H(s)}{1+C(S)H(s)} \\ &= \frac{n(s)b(s)}{d(s)a(s)+n(s)b(s)} \end{aligned} \tag{2.1.3}$$

Let the denominator be

$$d(s)a(s) + n(s)b(s) = f(s) \tag{2.1.4}$$

where $f(s) = const.(s - p_1)(s - p_2)\cdots(s - p_{\gamma_0 + k_0})$ contains the desired closed-loop poles p_i, $i = 1, 2, \cdots, (\gamma_0 + k_0)$, where $\gamma_0 = \deg[d(s)]$, $k_0 = \deg[a(s)]$. This equation is known as the Diophantine equation where $a(s)$, $b(s)$ and $f(s)$ are given polynomials and $d(s)$ and $n(s)$ are unknown polynomials to be solved. The Diophantine equation in Eq. (2.1.4) can be solved using Sylvester matrix method [1]. Here we introduce an alternative method using an order reduction approach employing the Euclidean algorithm. We shall revisit the Euclidean algorithm first and then proceed on to the order reduction method.

2.2 Euclidean Algorithm

This section revisits the Euclidean algorithm given in [1]. Euclidean algorithm is considered to be one of the oldest nontrivial algorithm that has survived to the present

day [34]. Given two polynomials $a_0(s)$ and $b_0(s)$ with $\deg[b_0(s)] < \deg[a_0(s)]$ there exists a unique quotient polynomial $q_0(s)$ and a unique remainder polynomial $r_0(s)$ such that

$$a_0(s) = q_0(s)b_0(s) + r_0(s) , \qquad \deg[r_0(s)] < \deg[b_0(s)] \qquad (2.2.1)$$

By successive use of the above polynomial division formula, we obtain

$$a_0(s) = q_0(s)b_0(s) + r_0(s) , \qquad k_2 < k_1$$

$$a_1(s) = q_1(s)b_1(s) + r_1(s) , \qquad k_3 < k_2$$

$$a_2(s) = q_2(s)b_2(s) + r_2(s) , \qquad k_4 < k_3 \qquad (2.2.2)$$

$$\vdots \qquad\qquad\qquad\qquad \vdots$$

$$a_{m-1}(s) = q_{m-1}(s)b_{m-1}(s) + r_{m-1}(s) , \qquad k_{m+1} < k_m$$

$$a_m(s) = q_m(s)b_m(s) + r_m(s) , \qquad \deg[r_m(s)] < k_{m+1}$$

where $a_i(s) = b_{i-1}(s)$ and $b_i(s) = r_{i-1}(s)$, $i = 1, \cdots, m$; $k_j = \deg[b_{j-1}(s)]$,

$j = 1, 2, \cdots, m+1$.

The algorithm stops when the remainder $r_m(s) = 0$ or $\deg[r_m(s)] = 0$. In the former case, $r_{m-1}(s)$ will be the greatest common divisor of $a_0(s)$ and $b_0(s)$. In the latter case, $a_0(s)$ and $b_0(s)$ are coprime.

2.3 Order Reduction of Diophantine Equation

Consider Eq. (2.1.4). Rename the polynomials as

$$a(s) = a_0(s) \qquad\qquad (2.3.1a)$$

$$b(s) = b_0(s) \qquad\qquad (2.3.1b)$$

8

$$d(s) = d_0(s) \qquad\qquad (2.3.1c)$$

$$n(s) = n_0(s) \qquad\qquad (2.3.1d)$$

$$f(s) = f_0(s) \qquad\qquad (2.3.1e)$$

Then, we have

$$d_0(s)a_0(s) + n_0(s)b_0(s) = f_0(s) \qquad\qquad (2.3.2)$$

where subscript "0" is used to indicate the step number in the recursive process that follows. It is assumed that $a_0(s)$ and $b_0(s)$ are coprime.

To go through a general discussion, let the degrees of the respective polynomials satisfy the following conditions:

$$\deg[a_0(s)] = k_0 \qquad\qquad (2.3.3a)$$

$$\deg[b_0(s)] = k_1, \qquad\qquad k_1 < k_0 \text{ (strictly proper plant)} \qquad\qquad (2.3.3b)$$

$$\deg[d_0(s)] \doteq \gamma_0, \qquad\qquad \gamma_0 \geq k_0 - 1 \text{ (proper controller)} \qquad\qquad (2.3.3c)$$

$$\deg[n_0(s)] = k_0 - 1 \qquad\qquad (2.3.3d)$$

$$\deg[f_0(s)] = \gamma_0 + k_0 \qquad\qquad (2.3.3e)$$

The choice of these degrees not only ensure a proper controller, but also will yield an equal number of equations as unknowns as can be seen below.

Let the polynomials be

$$a_0(s) = a_{0,0}s^{k_0} + a_{0,1}s^{k_0-1} + \cdots + a_{0,k_0-1}s + a_{0,k_0}, \qquad a_{0,0} \neq 0 \qquad (2.3.4a)$$

$$b_0(s) = b_{0,0}s^{k_1} + b_{0,1}s^{k_1-1} + \cdots + b_{0,k_1-1}s + b_{0,k_1}, \qquad k_1 < k_0 \qquad (2.3.4b)$$

$$d_0(s) = d_{0,0}s^{\gamma_0} + d_{0,1}s^{\gamma_0-1} + \cdots + d_{0,\gamma_0-1}s + d_{0,\gamma_0}, \qquad \gamma_0 \geq k_0 - 1 \qquad (2.3.4c)$$

$$n_0(s) = n_{0,0}s^{k_0-1} + n_{0,1}s^{k_0-2} + \cdots + n_{0,k_0-2}s + n_{0,k_0-1} \qquad (2.3.4d)$$

Let $f_0(s)$ be any arbitrarily assigned polynomial of degree $(\gamma_0 + k_0)$ given as

$$f_0(s) = f_{0,0}s^{\gamma_0+k_0} + f_{0,1}s^{\gamma_0+k_0-1} + \cdots + f_{0,\gamma_0+k_0-1}s + f_{0,\gamma_0+k_0} \qquad (2.3.5)$$

(We shall from now on refer to the degree of $f_0(s)$ as the order of the

Diophantine equation (2.3.2)).

Substituting Eqs. (2.3.4) and (2.3.5) into Eq. (2.3.2) and equating terms of like

powers in s yields the following matrix equation

$$(2.3.6)$$

10

The first $(\gamma_0 - k_1 + 1)$ coefficients of $d_0(s)$ can be solved by inverting a triangular matrix as

$$
\begin{bmatrix}
d_{0,0} \\
d_{0,1} \\
\vdots \\
d_{0,\gamma_0-k_1-1} \\
d_{0,\gamma_0-k_1}
\end{bmatrix}
=
\begin{bmatrix}
a_{0,0} & 0 & \cdots & \cdots & 0 \\
a_{0,1} & a_{0,0} & \ddots & \ddots & \vdots \\
a_{0,2} & a_{0,1} & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & 0 \\
a_{0,\gamma_0-k_1} & a_{0,\gamma_0-k_1-1} & \cdots & a_{0,1} & a_{0,0}
\end{bmatrix}^{-1}
\begin{bmatrix}
f_{0,0} \\
f_{0,1} \\
\vdots \\
f_{0,\gamma_0-k_1-1} \\
f_{0,\gamma_0-k_1}
\end{bmatrix}
\tag{2.3.7}
$$

Let $d_0(s)$ be split into a "known" leading part $d_{0H}(s)$ consisting of these coefficients and an "unknown" remaining part $d_{0L}(s)$ which is to be determined later such that

$$d_0(s) = d_{0H}(s) + d_{0L}(s) \tag{2.3.8a}$$

where

$$d_{0H}(s) = d_{0,0}s^{\gamma_0} + d_{0,1}s^{\gamma_0-1} + \cdots + d_{0,\gamma_0-k_1}s^{k_1} \tag{2.3.8b}$$

$$d_{0L}(p) = d_{0,\gamma_0-k_1+1}s^{k_1-1} + d_{0,\gamma_0-k_1+2}s^{k_1-2} + \cdots + d_{0,\gamma_0-1}s + d_{0,\gamma_0} \tag{2.3.8c}$$

Substituting Eq. (2.3.8a) into Eq. (2.3.2) yields

$$[d_{0H}(s) + d_{0L}(s)]a_0(s) + n_0(s)b_0(s) = f_0(s) \tag{2.3.9}$$

Moving all the known parts in Eq. (2.3.9) to the right hand-side gives

$$d_{0L}(s)a_0(p) + n_0(s)b_0(s) = f_0(s) - d_{0H}(s)a_0(s) \tag{2.3.10}$$

Denote the right hand-side as

$$
\begin{aligned}
f_1(s) &= f_0(s) - d_{0H}(s)a_0(s) \\
&= f_{1,0}s^{k_0+k_1-1} + f_{1,1}s^{k_0+k_1-2} + \cdots + f_{1,k_0+k_1-2}s + f_{1,k_0+k_1-1}
\end{aligned}
\tag{2.3.11}
$$

11

The coefficients of $f_1(s)$ can be found by substituting Eqs. (2.3.4a), (2.3.5) and (2.3.8b) into (2.3.11) so that we have

$$
\begin{bmatrix} f_{1,0} \\ f_{1,1} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f_{1,k_0+k_1-1} \end{bmatrix} = \begin{bmatrix} f_{0,\gamma_0-k_1+1} \\ f_{0,\gamma_0-k_1+2} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f_{0,\gamma_0+k_0} \end{bmatrix} - \begin{bmatrix} a_{0,\gamma_0-k_1+1} & a_{0,\gamma_0-k_1} & \cdots & a_{0,1} \\ a_{0,\gamma_0-k_1+2} & a_{0,\gamma_0-k_1+1} & \ddots & a_{0,2} \\ \vdots & \ddots & \ddots & \vdots \\ a_{0,k_0} & \ddots & \ddots & \vdots \\ 0 & a_{0,k_0} & \ddots & \vdots \\ \vdots & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{0,k_0} \\ \vdots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix} \begin{bmatrix} d_{0,0} \\ d_{0,1} \\ \vdots \\ d_{0,\gamma_0-k_1} \end{bmatrix}
\tag{2.3.12}
$$

With Eq. (2.3.11), Eq. (2.3.10) now becomes

$$
d_{0L}(s)a_0(s) + n_0(s)b_0(s) = f_1(s)
\tag{2.3.13}
$$

Substituting the Euclidean algorithm Eq. (2.2.1) into Eq. (2.3.13) gives

$$
d_{0L}(s)[q_0(s)b_0(s) + r_0(s)] + n_0(s)b_0(s) = f_1(s)
\tag{2.3.14}
$$

Grouping terms containing $b_0(s)$ yields

$$
[d_{0L}(s)q_0(s) + n_0(s)]b_0(s) + d_{0L}(s)r_0(s) = f_1(s)
\tag{2.3.15}
$$

Renaming the polynomials as

$$
a_1(s) = b_0(s)
\tag{2.3.16a}
$$

$$
b_1(s) = r_0(s)
\tag{2.3.16b}
$$

$$
d_1(s) = d_{0L}(s)q_0(s) + n_0(s)
\tag{2.3.16c}
$$

$$
n_1(s) = d_{0L}(s)
\tag{2.3.16d}
$$

gives rise to a new Diophantine equation

12

$$d_1(s)a_1(s) + n_1(s)b_1(s) = f_1(s) \tag{2.3.17}$$

where the degrees of the respective polynomials are

$$\deg[a_1(s)] = k_1 \tag{2.3.18a}$$

$$\deg[b_1(s)] = k_2, \qquad k_2 < k_1 \tag{2.3.18b}$$

$$\deg[d_1(s)] \doteq \gamma_1 = k_0 - 1 \tag{2.3.18c}$$

$$\deg[n_1(s)] = k_1 - 1 \tag{2.3.18d}$$

$$\deg[f_1(s)] = k_0 + k_1 - 1 \tag{2.3.18e}$$

Notice that they satisfy the same degree requirements as in Eq. (2.3.3). Consequently, the new Diophantine equation (2.3.17) is of lower order than the original one in Eq. (2.3.2). That is, $f_1(s)$ is of lower degree than $f_0(s)$, in this case by

$(\gamma_0 - k_1 + 1)$.

Repeating the above order reduction process one more time will yield third Diophantine equation:

$$d_2(s)a_2(s) + n_2(s)b_2(s) = f_2(s) \tag{2.3.19}$$

Continuing on with this process to a total number of $m + 1$ steps, where m is the number of steps in the Euclidean algorithm in Eq. (2.2.2), we will obtain a sequence of Diophantine equations of successively lower orders as

$$d_i(s)a_i(s) + n_i(s)b_i(s) = f_i(s), \quad \deg[f_i(s)] = \gamma_i + k_i \ , \quad i = 0, 1, \cdots, m \tag{2.3.20}$$

where

$$a_j(s) = b_{j-1}(s), \qquad\qquad\qquad j = 1, \cdots, m \tag{2.3.21a}$$

13

$$b_j(s) = r_{j-1}(s), \qquad\qquad j = 1, \cdots, m \qquad\qquad (2.3.21b)$$

$$d_j(s) = [d_{j-1}(s) - d_{(j-1)H}(s)]q_{j-1}(s) + n_{j-1}(s), \qquad j = 1, \cdots, m \qquad (2.3.21c)$$

$$n_j(s) = d_{j-1}(s) - d_{(j-1)H}(s), \qquad\qquad j = 1, \cdots, m \qquad\qquad (2.3.21d)$$

$$f_j(s) = f_{j-1}(s) - d_{(j-1)H}(s)a_{j-1}(s), \qquad\qquad j = 1, \cdots, m \qquad\qquad (2.3.21e)$$

$$d_{iH}(s) = d_{i,0}s^{\gamma_i} + d_{i,1}s^{\gamma_i-1} + \cdots + d_{i,\gamma_i-k_{i+1}}s^{k_{i+1}}, \qquad i = 0, 1, \cdots, m \qquad (2.3.21f)$$

The coefficients of $d_{iH}(s)$, $i = 0, 1, \cdots, m$, are obtained from

$$
\begin{bmatrix}
d_{i,0} \\
d_{i,1} \\
\vdots \\
d_{i,\gamma_i-k_{i+1}-1} \\
d_{i,\gamma_i-k_{i+1}}
\end{bmatrix}
=
\begin{bmatrix}
a_{i,0} & 0 & \cdots & \cdots & 0 \\
a_{i,1} & a_{i,0} & \ddots & \ddots & \vdots \\
\vdots & a_{i,1} & \ddots & \ddots & \vdots \\
\vdots & & \ddots & \ddots & 0 \\
a_{i,\gamma_i-k_{i+1}} & a_{i,\gamma_i-k_{i+1}-1} & \cdots & a_{i,1} & a_{i,0}
\end{bmatrix}^{-1}
\begin{bmatrix}
f_{i,0} \\
f_{i,1} \\
\vdots \\
f_{i,\gamma_i-k_{i+1}-1} \\
f_{i,\gamma_i-k_{i+1}}
\end{bmatrix}
\qquad (2.3.22)
$$

The degrees of the respective polynomials in Eq. (2.3.21) are given by

$$\deg[a_j(s)] = k_j \qquad\qquad\qquad (2.3.23a)$$

$$\deg[b_j(s)] = k_{j+1}, \qquad k_{j+1} < k_j \qquad\qquad (2.3.23b)$$

$$\deg[d_j(s)] \doteq \gamma_j = k_{j-1} - 1 \qquad\qquad (2.3.23c)$$

$$\deg[n_j(s)] = k_j - 1 \qquad\qquad (2.3.23d)$$

$$\deg[f_j(s)] = k_{j-1} + k_j - 1 \qquad\qquad (2.3.23e)$$

$$j = 1, \cdots, m$$

We now turn to the backward process starting with known pair, $d_m(s)$ and

$n_m(s)$. The objective here is to determine the original pair, $d_0(s)$ and $n_0(s)$. From Eqs.

(2.3.21d) and (2.3.21f), we have

$$d_{m-1}(s) = d_{(m-1)H}(s) + n_m(s) \tag{2.3.24a}$$

The polynomial $n_{m-1}(s)$ can be solved from Eqs. (2.3.21c) and (2.3.21d) as

$$n_{m-1}(s) = d_m(s) - n_m(s)q_{m-1}(s) \tag{2.3.24b}$$

Continuing on in this way, with the help of Eqs. (2.3.21c), (2.3.21d) and (2.3.21f), we have the following backward recursive formulas:

$$d_{i-1}(s) = d_{(i-1)H}(s) + n_i(s), \qquad i = 1, \cdots, m \tag{2.3.25a}$$

$$n_{i-1}(s) = d_i(s) - n_i(s)q_{i-1}(s), \qquad i = 1, \cdots, m \tag{2.3.25b}$$

2.4 Example of Time-Invariant Case

Consider a Diophantine equation (2.3.2) with the following polynomials

$$a_0(s) = s^3 + 3s^2 + 4s + 3$$

$$b_0(s) = s^2 + s + 1$$

$$d_0(s) = d_{0,0}s^3 + d_{0,1}s^2 + d_{0,2}s + d_{0,3}$$

$$n_0(s) = n_{0,0}s^2 + n_{0,1}s + n_{0,2}$$

Selecting all six closed-loop poles at $s = -1$ leads to

$$f_0(s) = (s+1)^6$$
$$= s^6 + 6s^5 + 15s^4 + 20s^3 + 15s^2 + 6s + 1$$

Applying the Euclidean algorithm Eq. (2.2.2) on $a_0(s)$ and $b_0(s)$ yields

$$q_0(s) = s + 2$$

15

$$r_0(s) = s + 1$$

A second application of the Euclidean algorithm on $a_1(s)$ and $b_1(s)$ gives

$$q_1(s) = s$$

$$r_1(s) = 1$$

The Euclidean algorithm stops here since $\deg[r_1(s)] = 0$. Next we proceed to reduce the order of the Diophantine equation.

2.4.1 Forward Algorithm

Step 0:

From Eq. (2.3.22), we can solve for the two leading coefficients of $d_0(s)$ as

$$\begin{bmatrix} d_{0,0} \\ d_{0,1} \end{bmatrix} = \begin{bmatrix} a_{0,0} & 0 \\ a_{0,1} & a_{0,0} \end{bmatrix}^{-1} \begin{bmatrix} f_{0,0} \\ f_{0,1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 6 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

Then, $d_0(s)$ can be split into a "known" part and an "unknown" part as

$$d_{0H}(s) = s^3 + 3s^2$$

$$d_{0L}(s) = d_{0,2}s + d_{0,3}$$

From Eqs. (2.3.20) and (2.3.21), with $i = 1$, the following reduced order Diophantine equation is obtained

$$d_1(s)a_1(s) + n_1(s)b_1(s) = f_1(s)$$

where

16

$a_1(s) = s^2 + s + 1$

$b_1(s) = s + 1$

$f_1(s) = 2s^4 + 5s^3 + 6s^2 + 6s + 1$

Step 1:

From Eq. (2.3.22), we can solve for the two leading coefficients in $d_1(s)$ as

$$\begin{bmatrix} d_{1,0} \\ d_{1,1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 2 \\ 5 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

So that, $d_1(s)$ will be split into a "known" part and an "unknown" part as

$d_{1H}(s) = 2s^2 + 3s$

$d_{1L}(s) = d_{1,2}$

From Eqs. (2.3.20) and (2.3.21), with $i = 2$, another reduced order Diophantine equation is generated

$$d_2(s)a_2(s) + n_2(s)b_2(s) = f_2(s) \tag{2.4.1}$$

where

$a_2(s) = s + 1$

$b_2(s) = 1$

$f_2(s) = s^2 + 3s + 1$

Step 2:

From Eq. (2.3.22) we can obtain $d_{2,0}$, $d_{2,1}$ and $n_{2,0}$ as follows

17

$$\begin{bmatrix} d_{2,0} \\ d_{2,1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Notice that, in this final step, $d_2(s) = d_{2H}(s) = s + 2$. With known $d_2(s)$, $n_2(s)$ follows from the Diophantine equation (2.4.1) as

$$n_{2,0} = -1$$

2.4.2 Backward Algorithm

With known $d_2(s)$ and $n_2(s)$, $d_1(s)$ and $n_1(s)$ can be obtained from Eq. (2.3.25) as

$$d_1(s) = 2s^2 + 3s - 1$$

$$n_1(s) = 2s + 2$$

Similarly, with $d_1(s)$ and $n_1(s)$ solved, $d_0(s)$ and $n_0(s)$ are found from Eq. (2.3.25) as

$$d_0(s) = s^3 + 3s^2 + 2s + 2$$

$$n_0(s) = -3s - 5$$

which is the solution of our original Diophantine equation.

CHAPTER 3

ORDER REDUCTION OF TIME-VARYING DIOPHANTINE EQUATION USING EUCLIDEAN ALGORITHM

Now we extend the approach described in the previous chapter to time-varying case.

3.1 Time-Varying Pole-Placement Feedback Design

Pole-placement in a closed-loop structure using output feedback can be carried out by means of solving the Diophantine equation. The poles of the overall transfer function operator are assigned in order to meet certain given performance requirements.

Consider a plant described by a linear time-varying differential equation

$$A(p)y(t) = B(p)u(t) \qquad (3.1.1)$$

where

$$A(p) = a_0(t)p^{k_0} + a_1(t)p^{k_0-1} + \cdots + a_i(t)p^{k_0-i} + \cdots + a_{k_0}(t), \qquad (3.1.2)$$

$$a_0(t) \neq 0, t_1 < t < t_2$$

$$B(p) = b_0(t)p^{k_1} + b_1(t)p^{k_1-1} + \cdots + b_j(t)p^{k_1-j} + \cdots + b_{k_1}(t), \ \ k_1 < k_0 \qquad (3.1.3)$$

and p is the differential operator $\dfrac{d}{dt}$. The coefficients a_i $(i = 0, \cdots, k_0)$ and

b_j $(j = 0, \cdots, k_1)$ are continuously differentiable functions of time. In the sequel, the

argument t in the coefficients a_i and b_j will be dropped for brevity.

19

A feedback control system for the plant can be implemented as shown in Figure

3.1 where r is the reference input. This scheme can be viewed as the combined

observer-controller feedback configuration given in [1] (pp.298), which is now

extended to the time-varying case. The objective here is to design H_u and H_y so that

the closed-loop poles would be assigned.

PLAN
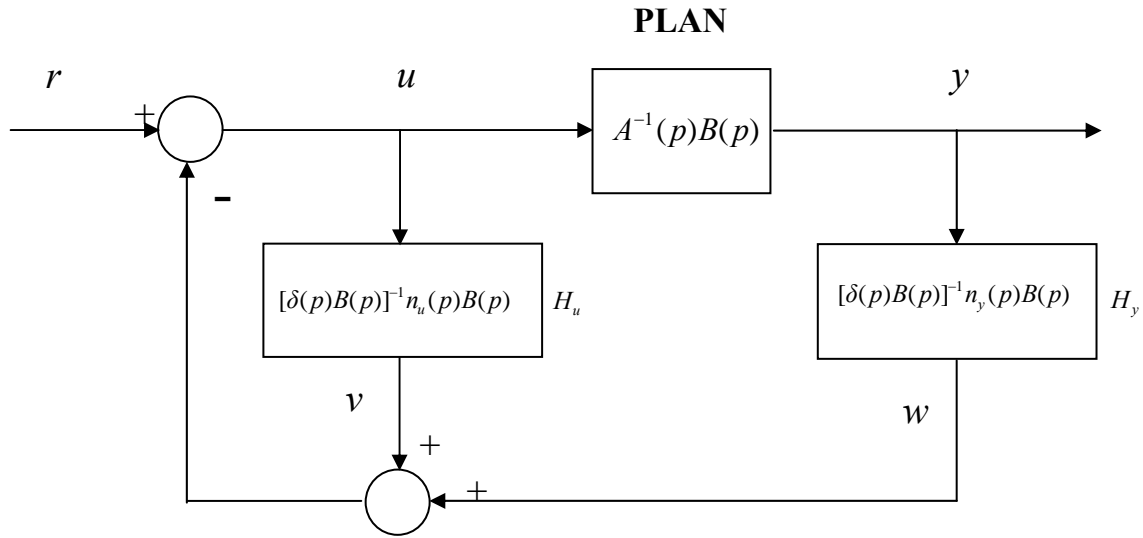
Figure 3.1. Block diagram of combined observer-controller

In Figure 3.1, the blocks H_u and H_y can be represented as polynomial operator

equations, respectively, as

$$\delta(p)B(p)v = n_u(p)B(p)u \tag{3.1.4}$$

$$\delta(p)B(p)w = n_y(p)B(p)y \tag{3.1.5}$$

where

$$\delta(p) = \delta_0(p - p_{o1})(p - p_{o2})\cdots(p - p_{o\sigma_0})$$
$$= \delta_0 p^{\sigma_0} + \delta_1 p^{\sigma_0-1} + \cdots + \delta_{\sigma_0-1}p + \delta_{\sigma_0} \tag{3.1.6}$$

The constants p_{oi}, $(i = 1, \cdots, \sigma_0)$, are the assigned "observer" poles.

Let

$$n_u(p) = \gamma_0 p^{\sigma_0-1} + \cdots + \gamma_{\sigma_0-2}p + \gamma_{\sigma_0-1} \tag{3.1.7}$$

$$n_y(p) = \beta_0 p^{k_0-1} + \cdots + \beta_{k_0-2}p + \beta_{k_0-1}, \qquad k_0 - 1 \le \sigma_0 \tag{3.1.8}$$

Then the overall feedback system equation can be derived as follows:

Summing up Eq. (3.1.4) and Eq. (3.1.5) results in

$$\delta(p)B(p)[v + w] = n_u(p)B(p)u + n_y(p)B(p)y \tag{3.1.9}$$

From Figure 3.1, we have $v + w = r - u$. Eq. (3.1.9) then becomes

$$\delta(p)B(p)[r - u] = n_u(p)B(p)u + n_y(p)B(p)y \tag{3.1.10}$$

Eliminating u from Eq. (3.1.1) and Eq. (3.1.10) yields the overall closed-loop

equation

$$\left\{ [\delta(p) + n_u(p)]A(p) + n_y(p)B(p) \right\} y = \delta(p)B(p)r \tag{3.1.11}$$

Let the left-hand-side term be

$$\left\{ [\delta(p) + n_u(p)]A(p) + n_y(p)B(p) \right\} = \delta(p)\alpha(p) \tag{3.1.12}$$

where

$$\alpha(p) = \alpha_0(p - p_{c1})(p - p_{c2})\cdots(p - p_{ck_0})$$
$$= \alpha_0 p^{k_0} + \alpha_1 p^{k_0-1} + \cdots + \alpha_{k_0-1}p + \alpha_{k_0} \quad,$$

The constants p_{ci} $(i = 1, \cdots, k_0)$ are the assigned "controller" poles.

Then, substituting Eq. (3.1.12) into Eq. (3.1.11) gives

21

$$\delta(p)\alpha(p)y = \delta(p)B(p)r \tag{3.1.13a}$$

or

$$\delta(p)[\alpha(p)y - B(p)r] = 0 \tag{3.1.13b}$$

Integrating Eq. (3.1.13b) leads to

$$\begin{aligned} \alpha(p)y &= B(p)r + \zeta(t) \\ &\cong B(p)r \end{aligned} \tag{3.1.14}$$

where $\zeta(t)$ is a transient signal and decays away rapidly by applying differential polynomial $\delta(p)$.

The equivalent overall feedback system is depicted in Figure 3.2.

$$\tilde{r} \longrightarrow \boxed{B(p)^{-1}} \xrightarrow{\ r\ } \boxed{\alpha(p)} \xrightarrow{\ y\ }$$
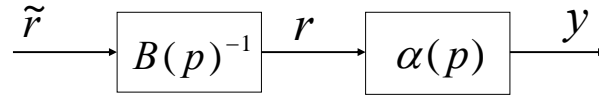
Figure 3.2. Block diagram of equivalent overall combined observer-controller

Remarks:

(1)   With fixed closed-loop poles, p_{ci}, $\alpha(p)$ is a time-invariant polynomial, which governs the behavior of the closed-loop system.

(2)   The combined observer-controller approach described above requires no separate knowledge of the controllability or observability of the plant, and no state measurement is needed. The only requirement is that $A(p)$ and $B(p)$ are right coprime [22].

With respect to Eq. (3.1.12), let

$$D(p) = \delta(p) + n_u(p)$$
$$= d_0 p^{\sigma_0} + d_1 p^{\sigma_0 - 1} + \cdots + d_{\sigma_0 - 1} p + d_{\sigma_0} \qquad (3.1.15a)$$

$$N(p) = n_y(p)$$
$$= n_0 p^{k_0 - 1} + \cdots + n_{k_0 - 2} p + n_{k_0 - 1} \qquad (3.1.15b)$$

$$F(p) = \delta(p)\alpha(p)$$
$$= f_0 p^{\sigma_0 + k_0} + f_1 p^{\sigma_0 + k_0 - 1} + f_2 p^{\sigma_0 + k_0 - 2} + \cdots + f_{\sigma_0 + k_0 - 1} p + f_{\sigma_0 + k_0} \qquad (3.1.15c)$$

Then, Eq. (3.1.12) becomes

$$D(p)A(p) + N(p)B(p) = F(p). \qquad (3.1.16)$$

Thus, our design goal is reduced to the problem of solving the Diophantine equation (3.1.16). With given $A(p)$, $B(p)$ and $F(p)$, $D(p)$ and $N(p)$ are to be solved.

3.2 Time-Varying Division Algorithm

Consider a pair of polynomials $A(s)$ and $B(s)$ in the Laplace operator s. In this time-invariant case, it is well known [1][23] that the division of $A(s)$ by $B(s)$ uniquely defines another pair of polynomials, the remainder $R(s)$ and the quotient $Q(s)$, such that

$$A(s) = Q(s)B(s) + R(s) \qquad (3.2.1)$$

Extending this to the time-varying case, Eq. (3.2.1) becomes

$$A(p) = Q(p)B(p) + R(p) \qquad (3.2.2)$$

Let the polynomials be given by

23

$$A(p) = a_0 p^{k_0} + a_1 p^{k_0-1} + \cdots + a_{k_0-1} p + a_{k_0}, \quad a_0(t) \neq 0 \quad \text{for} \quad t_1 < t < t_2 \quad \text{(3.2.3a)}$$

$$B(p) = b_0 p^{k_1} + b_1 p^{k_1-1} + \cdots + b_{k_1-1} p + b_{k_1}, \quad b_0(t) \neq 0 \quad \text{for} \quad t_1 < t < t_2 \quad \text{(3.2.3b)}$$

$$R(p) = r_0 p^{k_1-1} + r_1 p^{k_1-2} + \cdots + r_{k_1-2} p + r_{k_1-1} \quad \text{(3.2.3c)}$$

$$Q(p) = q_0 p^{k_0-k_1} + q_1 p^{k_0-k_1-1} + \cdots + q_{k_0-k_1-1} p + q_{k_0-k_1}, \quad k_0 > k_1 \quad \text{(3.2.3d)}$$

Substituting Eq. (3.2.3) into Eq. (3.2.2) and equating terms of like powers in p yields the following square triangular systems of equations:

$$
\begin{bmatrix}
a_0 \\
a_1 \\
a_2 \\
\vdots \\
\vdots \\
a_{k_0-k_1} \\
\text{---} \\
a_{k_0-k_1+1} \\
\vdots \\
\vdots \\
\vdots \\
a_{k_0-1} \\
a_{k_0}
\end{bmatrix}
=
\begin{bmatrix}
b_0 & 0 & 0 & \cdots & \cdots & 0 & | & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\
b_1+\bar{b}_{1,0} & b_0 & 0 & \ddots & \ddots & \vdots & | & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
b_2+\bar{b}_{2,0} & b_1+\bar{b}_{2,1} & b_0 & \ddots & \ddots & \vdots & | & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & | & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & 0 & | & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & b_0 & | & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\
\text{---} & \text{---} & \text{---} & - & - & - & -- & + & - & - & - & - & - & - \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & b_1 & | & 1 & 0 & \cdots & \cdots & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & b_2 & | & 0 & 1 & \ddots & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & | & 0 & 0 & \ddots & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots & | & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\bar{b}_{k_0-1,0} & \bar{b}_{k_0-1,1} & \ddots & \ddots & \ddots & b_{k_1-1} & | & 0 & \ddots & \ddots & \ddots & 1 & 0 \\
\bar{b}_{k_0,0} & \bar{b}_{k_0,1} & \cdots & \cdots & \cdots & b_{k_1} & | & 0 & \cdots & \cdots & \cdots & 0 & 1
\end{bmatrix}
\begin{bmatrix}
q_0 \\
q_1 \\
q_2 \\
\vdots \\
\vdots \\
q_{k_0-k_1} \\
\text{---} \\
r_0 \\
\vdots \\
\vdots \\
\vdots \\
\vdots \\
r_{k_1-1}
\end{bmatrix}
$$

$$\text{(3.2.4)}$$

The terms $\bar{b}_{j+i+l,j}$ in Eq. (3.2.4) exist only in the time-varying case and are defined as

$$\bar{b}_{j+i+l,j} = \sum_{i=1}^{k_0-k_1-j} \binom{k_0-k_1-j}{i} \sum_{l=0}^{k_1} b_l^{(i)}, \quad j = 0, 1, \cdots, (k_0-k_1-1) \quad \text{(3.2.5)}$$

24

where $\begin{pmatrix} k_0 - k_1 - j \\ i \end{pmatrix}$ is the combinatorial symbol.

From Eq. (3.2.4), the unknown coefficients q's and r's can be found uniquely if the matrix is nonsingular. This is the case since $b_0(t) \neq 0$, $t_1 < t < t_2$.

We shall refer to the degree of $R(p)$ as

$$\deg[R(p)] \doteq k_2 \qquad\qquad (3.2.6)$$

if the coefficient $r_{k_1-k_2-1}(t) \neq 0$, $t_1 < t < t_2$, $(k_2 = 0, 1, \cdots, k_1 - 1)$, and that all coefficients previous to it are identically equal to zero.

3.3 Time-Varying Euclidean Algorithm

This section revisits the Euclidean algorithm given in [1]. Euclidean algorithm is considered to be one of the oldest nontrivial algorithm that has survived to the present day [34]. Given two polynomials $A_0(p)$ and $B_0(p)$ with $\deg[B_0(p)] < \deg[A_0(p)]$, using the division algorithm in Section 3, we have

$$A_0(p) = Q_0(p)B_0(p) + R_0(p) , \qquad \deg[R_0(p)] < \deg[B_0(p)] \qquad (3.3.1)$$

By successive use of the polynomial division algorithm, we obtain, for $t_1 < t < t_2$,

$$A_0(p) = Q_0(p)B_0(p) + R_0(p), \qquad k_2 < k_1$$

$$A_1(p) = Q_1(p)B_1(p) + R_1(p) , \qquad k_3 < k_2$$

25

$$A_2(p) = Q_2(p)B_2(p) + R_2(p) , \qquad k_4 < k_3 \qquad\qquad (3.3.2)$$

$$\vdots \qquad\qquad\qquad\qquad \vdots$$

$$A_{m-1}(p) = Q_{m-1}(p)B_{m-1}(p) + R_{m-1}(p) , \quad k_{m+1} < k_m$$

$$A_m(p) = Q_m(p)B_m(p) + R_m(p), \qquad \deg[R_m(p)] < k_{m+1}$$

where $A_i(p) = B_{i-1}(p)$ and $B_i(p) = R_{i-1}(p)$, $i = 1, \cdots, m$; $k_j = \deg[B_{j-1}(p)]$,

$j = 1, 2, \cdots, m+1$.

The algorithm stops when the remainder $R_m(p) = 0$ or, when $R_m(p) \neq 0$,

$\deg[R_m(p)] = 0$. In the former case, $R_{m-1}(p)$ will be the greatest common right divisor

of $A_0(p)$ and $B_0(p)$. In the latter case, $A_0(p)$ and $B_0(p)$ are right coprime. The

proof is similar to the time-invariant case and will be omitted.

3.4 Order Reduction of Time-Varying Diophantine Equation

Consider Eq. (3.1.16). Rename the polynomials as

$$A(p) = A_0(p) \qquad\qquad\qquad\qquad\qquad\qquad\qquad (3.4.1a)$$

$$B(p) = B_0(p) \qquad\qquad\qquad\qquad\qquad\qquad\qquad (3.4.1b)$$

$$D(p) = D_0(p) \qquad\qquad\qquad\qquad\qquad\qquad\qquad (3.4.1c)$$

$$N(p) = N_0(p) \qquad\qquad\qquad\qquad\qquad\qquad\qquad (3.4.1d)$$

$$F(p) = F_0(p) \qquad\qquad\qquad\qquad\qquad\qquad\qquad (3.4.1e)$$

where

$$A_0(p) = a_{0,0}p^{k_0} + a_{0,1}p^{k_0-1} + \cdots + a_{0,k_0-1}p + a_{0,k_0}, \quad a_{0,0} \neq 0, \ t_1 < t < t_2 \qquad (3.4.2a)$$

$$B_0(p) = b_{0,0}p^{k_1} + b_{0,1}p^{k_1-1} + \cdots + b_{0,k_1-1}p + b_{0,k_1}, \quad k_1 < k_0, \ t_1 < t < t_2 \qquad (3.4.2b)$$

26

$$D_0(p) = d_{0,0}p^{\sigma_0} + d_{0,1}p^{\sigma_0-1} + \cdots + d_{0,\sigma_0-1}p + d_{0,\sigma_0}, \qquad \sigma_0 \geq k_0 - 1 \quad (3.4.2c)$$

$$N_0(p) = n_{0,0}p^{k_0-1} + n_{0,1}p^{k_0-2} + \cdots + n_{0,k_0-2}p + n_{0,k_0-1} \qquad (3.4.2d)$$

$$F_0(p) = f_{0,0}p^{\sigma_0+k_0} + f_{0,1}p^{\sigma_0+k_0-1} + \cdots + f_{0,\sigma_0+k_0-1}p + f_{0,\sigma_0+k_0} \qquad (3.4.2e)$$

Then, we have

$$D_0(p)A_0(p) + N_0(p)B_0(p) = F_0(p) \qquad (3.4.3)$$

The subscript "0" is used to indicate the step number in the recursive process that follows. It is assumed that $A_0(p)$ and $B_0(p)$ are right coprime. We shall from now on refer to the degree of $F_0(p)$ as the order of the Diophantine equation (3.4.3).

Our discussion is divided into two parts, a forward algorithm and a backward algorithm.

3.4.1 Forward Algorithm

We start with noticing that the degrees of the respective polynomials in Eq. (3.4.1) satisfy the following conditions:

$$\deg[A_0(p)] = k_0 \qquad (3.4.4a)$$

$$\deg[B_0(p)] = k_1, \qquad\qquad k_1 < k_0 \text{ (because of strictly proper plant)} \quad (3.4.4b)$$

$$\deg[D_0(p)] \doteq \sigma_0, \qquad\qquad \sigma_0 \geq k_0 - 1 \text{ (because of proper controller)} \quad (3.4.4c)$$

$$\deg[N_0(p)] = k_0 - 1 \qquad (3.4.4d)$$

$$\deg[F_0(p)] = \sigma_0 + k_0 - 1 \qquad (3.4.4e)$$

These degrees not only ensure a proper controller, but also will yield an equal number of equations as unknowns as can be seen below.

27

Substituting Eq. (3.4.2) into Eq. (3.4.3) and equating terms of like powers in p yields the following block-triangular matrix equation

$$
\left[\begin{array}{c|c}
\mathbf{S}_{11} & \mathbf{0} \\
\hline
\mathbf{S}_{21} & \mathbf{S}_{22}
\end{array}\right]\left[\begin{array}{c}
\mathbf{v}_H \\
\hline
\mathbf{v}_L
\end{array}\right]=\left[\begin{array}{c}
\mathbf{f}_H \\
\hline
\mathbf{f}_L
\end{array}\right]
\tag{3.4.5}
$$

where

$$
\mathbf{S}_{11}=\left[\begin{array}{ccccc}
a_{0,0} & 0 & \cdots & \cdots & 0 \\
a_{0,1}+\bar{a}_{1,0} & a_{0,0} & \ddots & \ddots & \vdots \\
a_{0,2}+\bar{a}_{2,0} & a_{0,1}+\bar{a}_{2,1} & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & 0 \\
a_{0,\sigma_0-k_1}+\bar{a}_{\sigma_0-k_1,0} & a_{0,\sigma_0-k_1-1}+\bar{a}_{\sigma_0-k_1,1} & \cdots & a_{0,1}+\bar{a}_{\sigma_0-k_1,\sigma_0-k_1-1} & a_{0,0}
\end{array}\right]
\tag{3.4.6a}
$$

$$
\mathbf{S}_{21}=\left[\begin{array}{ccccc}
a_{0,\sigma_0-k_1+1}+\bar{a}_{\sigma_0-k_1+1,0} & a_{0,\sigma_0-k_1}+\bar{a}_{\sigma_0-k_1+1,1} & \cdots & \cdots & a_{0,1}+\bar{a}_{\sigma_0-k_1+1,\sigma_0-k_1} \\
a_{0,\sigma_0-k_1+2}+\bar{a}_{\sigma_0-k_1+2,0} & a_{0,\sigma_0-k_1+1}+\bar{a}_{\sigma_0-k_1+2,1} & \ddots & \ddots & a_{0,2}+\bar{a}_{\sigma_0-k_1+2,\sigma_0-k_1} \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
a_{0,k_0}+\bar{a}_{k_0,0} & \ddots & \ddots & \ddots & \vdots \\
\bar{a}_{k_0+1,0} & a_{0,k_0}+\bar{a}_{k_0+1,1} & \ddots & \ddots & \vdots \\
\vdots & \bar{a}_{k_0+1,1} & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & a_{0,k_0}+\bar{a}_{\sigma_0+k_0-k_1,\sigma_0-k_1} \\
\vdots & \ddots & \ddots & \ddots & \bar{a}_{\sigma_0+k_0-k_1+1,\sigma_0-k_1} \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
\bar{a}_{\sigma_0+k_0,0} & \bar{a}_{\sigma_0+k_0,1} & \cdots & \bar{a}_{\sigma_0+k_0,\sigma_0-k_1-1} & \bar{a}_{\sigma_0+k_0,\sigma_0-k_1}
\end{array}\right]
$$

$$
\tag{3.4.6b}
$$

$$\mathbf{S}_{22} = \begin{bmatrix} a_{0,0} & 0 & 0 & b_{0,0} & 0 & \cdots & 0 \\ a_{0,1}+\overline{a}_{\sigma_0-k_1+2,\sigma_0-k_1+1} & \ddots & 0 & b_{0,1}+\overline{b}_{1,0} & b_{0,0} & \ddots & \vdots \\ a_{0,2}+\overline{a}_{\sigma_0-k_1+3,\sigma_0-k_1+1} & \ddots & a_{0,0} & b_{0,2}+\overline{b}_{2,0} & b_{0,1}+\overline{b}_{2,1} & \ddots & 0 \\ \vdots & \ddots & a_{0,1} & \vdots & \ddots & \ddots & b_{0,0} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & b_{0,1} \\ \vdots & \ddots & \vdots & b_{0,k_1}+\overline{b}_{k_1,0} & \ddots & \ddots & \vdots \\ a_{0,k_0}+\overline{a}_{\sigma_0+k_0-k_1,\sigma_0-k_1+1} & \ddots & \vdots & \overline{b}_{k_1+1,0} & b_{0,k_1}+\overline{b}_{k_1+1,1} & \ddots & \vdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\ \overline{a}_{\sigma_0+k_0,\sigma_0-k_1+1} & \cdots & a_{0,k_0} & b_{0,k_1}^{(k_0-1)} & \cdots & \cdots & b_{0,k_1} \end{bmatrix} \tag{3.4.6c}$$

$$\mathbf{v_H} = \begin{bmatrix} d_{0,0} & d_{0,1} & \cdots & d_{0,\sigma_0-k_1} \end{bmatrix}^T \tag{3.4.6d}$$

$$\mathbf{v_L} = \begin{bmatrix} d_{0,\sigma_0-k_1+1} & \cdots & d_{0,\sigma_0} & n_{0,0} & n_{0,1} & \cdots & n_{0,k_0-1} \end{bmatrix}^T \tag{3.4.6e}$$

$$\mathbf{f_H} = \begin{bmatrix} f_{0,0} & f_{0,1} & \cdots & f_{0,\sigma_0-k_1-1} & f_{0,\sigma_0-k_1} \end{bmatrix}^T \tag{3.4.6f}$$

$$\mathbf{f_L} = \begin{bmatrix} f_{0,\sigma_0-k_1+1} & f_{0,\sigma_0-k_1+2} & \cdots & f_{0,\sigma_0} & \cdots & f_{0,\sigma_0+k_0-1} & f_{0,\sigma_0+k_0} \end{bmatrix}^T \tag{3.4.6g}$$

The terms $\overline{a}_{j+i+l,j}$ and $\overline{b}_{j+i+l,j}$ in Eq. (3.4.6) will drop out in the time-invariant

case and are defined by

$$\overline{a}_{j+i+l,j} = \sum_{i=1}^{\sigma_0-j} \binom{\sigma_0-j}{i} \sum_{l=0}^{k_0} a_l^{(i)} , \qquad j = 0, 1, \cdots, (\sigma_0-1) \tag{3.4.7a}$$

$$\overline{b}_{j+i+l,j} = \sum_{i=1}^{k_0-1-j} \binom{k_0-1-j}{i} \sum_{l=0}^{k_1} b_l^{(i)} , \qquad j = 0, 1, \cdots, (k_1-2) \tag{3.4.7b}$$

Based on the above matrix, one can see that the first $(\sigma_0 - k_1 + 1)$ coefficients in

$D_0(p)$ can be solved by inverting a triangular matrix as

$$\mathbf{v_H} = \mathbf{S}_{11}^{-1} \mathbf{f_H} \tag{3.4.8}$$

Let $D_0(p)$ be split into a "known" leading part $D_{0H}(p)$ whose coefficients are found by Eq. (3.4.8) and an "unknown" remaining part $D_{0L}(p)$ which is to be found later such that

$$D_0(p) = D_{0H}(p) + D_{0L}(p) \tag{3.4.9}$$

where

$$D_{0H}(p) = d_{0,0}p^{\sigma_0} + d_{0,1}p^{\sigma_0-1} + \cdots + d_{0,\sigma_0-k_1}p^{k_1} \tag{3.4.10a}$$

$$D_{0L}(p) = d_{0,\sigma_0-k_1+1}p^{k_1-1} + d_{0,\sigma_0-k_1+2}p^{k_1-2} + \cdots + d_{0,\sigma_0-1}p + d_{0,\sigma_0} \tag{3.4.10b}$$

Substituting Eq. (3.4.10a) into Eq. (3.3.2) yields

$$[D_{0H}(p) + D_{0L}(p)]A_0(p) + N_0(p)B_0(p) = F_0(p) \tag{3.4.11}$$

Moving all known parts in Eq. (3.4.11) to the right hand-side gives

$$D_{0L}(p)A_0(p) + N_0(p)B_0(p) = F_0(p) - D_{0H}(p)A_0(p) \tag{3.4.12}$$

Denote the right hand-side as

$$\begin{aligned} F_1(p) &= F_0(p) - D_{0H}(p)A_0(p) \\ &= f_{1,0}p^{k_0+k_1-1} + f_{1,1}p^{k_0+k_1-2} + \cdots + f_{1,k_0+k_1-2}p + f_{1,k_0+k_1-1} \end{aligned} \tag{3.4.13}$$

The coefficients of $F_1(p)$ can be found by substituting Eqs. (3.4.2a), (3.4.3) and (3.4.10a) into (3.4.13) as

$$
\begin{bmatrix} f_{1,0} \\ f_{1,1} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f_{1,k_0+k_1-1} \end{bmatrix}
=
\begin{bmatrix} f_{0,\sigma_0-k_1+1} \\ f_{0,\sigma_0-k_1+2} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f_{0,\sigma_0+k_0} \end{bmatrix}
$$

$$
-
\begin{bmatrix}
a_{0,\sigma_0-k_1+1}+\overline{a}_{\sigma_0-k_1+1,0} & a_{0,\sigma_0-k_1}+\overline{a}_{\sigma_0-k_1+1,1} & \cdots & a_{0,1}+\overline{a}_{\sigma_0-k_1+1,\sigma_0-k_1} \\
a_{0,\sigma_0-k_1+2}+\overline{a}_{\sigma_0-k_1+2,0} & a_{0,\sigma_0-k_1+1}+\overline{a}_{\sigma_0-k_1+2,1} & \ddots & a_{0,2}+\overline{a}_{\sigma_0-k_1+2,\sigma_0-k_1} \\
\vdots & \ddots & \ddots & \vdots \\
a_{0,k_0}+\overline{a}_{k_0,0} & \ddots & \ddots & \vdots \\
\overline{a}_{k_0+1,0} & a_{0,k_0}+\overline{a}_{k_0+1,1} & \ddots & \vdots \\
\vdots & \overline{a}_{k_0+1,1} & \ddots & \vdots \\
\vdots & \ddots & \ddots & a_{0,k_0}+\overline{a}_{\sigma_0+k_0-k_1,\sigma_0-k_1} \\
\vdots & \ddots & \ddots & \overline{a}_{\sigma_0+k_0-k_1+1,\sigma_0-k_1} \\
\vdots & \ddots & \ddots & \vdots \\
\overline{a}_{\sigma_0+k_0,0} & \overline{a}_{\sigma_0+k_0,1} & \cdots & \overline{a}_{\sigma_0+k_0,\sigma_0-k_1}
\end{bmatrix}
\begin{bmatrix} d_{0,0} \\ d_{0,1} \\ \vdots \\ d_{0,\sigma_0-k_1} \end{bmatrix}
$$

$$\text{(3.4.14)}$$

With Eq. (3.4.13), Eq. (3.4.12) now becomes

$$D_{0L}(p)A_0(p)+N_0(p)B_0(p)=F_1(p) \tag{3.4.15}$$

Substituting the Euclidean algorithm Eq. (3.3.1) into Eq. (3.4.15) gives

$$D_{0L}(p)[Q_0(p)B_0(p)+R_0(p)]+N_0(p)B_0(p)=F_1(p) \tag{3.4.16}$$

Grouping terms containing $B_0(p)$ yields

$$[D_{0L}(p)Q_0(p)+N_0(p)]B_0(p)+D_{0L}(p)R_0(p)=F_1(p) \tag{3.4.17}$$

Renaming the polynomials as

$$A_1(p) = B_0(p) \tag{3.4.18a}$$

$$B_1(p) = R_0(p) \tag{3.4.18b}$$

$$D_1(p) = D_{0L}(p)Q_0(p) + N_0(p) \tag{3.4.18c}$$

$$N_1(p) = D_{0L}(p) \tag{3.4.18d}$$

gives rise to a new Diophantine equation

$$D_1(p)A_1(p) + N_1(p)B_1(p) = F_1(p) \tag{3.4.19}$$

The degrees of the respective polynomials are given by

$$\deg[A_1(p)] = k_1 \tag{3.4.20a}$$

$$\deg[B_1(p)] = k_2, \qquad k_2 < k_1 \tag{3.4.20b}$$

$$\deg[D_1(p)] \doteq \sigma_1 = k_0 - 1 \tag{3.4.20c}$$

$$\deg[N_1(p)] = k_1 - 1 \tag{3.4.20d}$$

$$\deg[F_1(p)] = k_0 + k_1 - 1 \tag{3.4.20e}$$

Notice that they satisfy the same degree requirements as in Eq. (3.4.3).

Consequently, the new Diophantine equation (3.4.19) is of lower order than that of the

original one in Eq. (3.4.3). That is, $F_1(p)$ is of lower degree than $F_0(p)$, specifically by

$(\sigma_0 - k_1 + 1)$.

Repeating the above order reduction process one more time will yield a third

Diophantine equation:

$$D_2(p)A_2(p) + N_2(p)B_2(p) = F_2(p) \tag{3.4.21}$$

Continuing on with this process to a total number of $m+1$ steps, where m is the number of steps in the Euclidean algorithm in Eq. (3.3.2), we will obtain a sequence of Diophantine equations of successively lower orders as

$$D_i(p)A_i(p) + N_i(p)B_i(p) = F_i(p), \tag{3.4.22}$$

$$\deg[F_i(p)] = \sigma_i + k_i, \qquad i = 0, 1, \cdots, m$$

where

$$A_j(p) = B_{j-1}(p), \qquad\qquad j = 1, \cdots, m \tag{3.4.23a}$$

$$B_j(p) = R_{j-1}(p), \qquad\qquad j = 1, \cdots, m \tag{3.4.23b}$$

$$\begin{aligned} D_j(p) &= [D_{j-1}(p) - D_{(j-1)H}(p)]Q_{j-1}(p) + N_{j-1}(p) \\ &= d_{j,0}p^{\sigma_j} + d_{j,1}p^{\sigma_j-1} + \cdots + d_{j,\sigma_j-1}p + d_{0,\sigma_j} \end{aligned}, \qquad j = 1, \cdots, m \tag{3.4.23c}$$

$$\begin{aligned} N_j(p) &= D_{j-1}(p) - D_{(j-1)H}(p) \\ &= n_{j,0}p^{k_j-1} + n_{j,1}p^{k_j-2} + \cdots + n_{j,k_j-2}p + n_{0,k_j-1} \end{aligned}, \qquad j = 1, \cdots, m \tag{3.4.23d}$$

$$\begin{aligned} F_j(p) &= F_{j-1}(p) - D_{(j-1)H}(p)A_{j-1}(p) \\ &= f_{j,0}p^{\sigma_j+k_j} + f_{j,1}p^{\sigma_j+k_j-1} + \cdots + f_{j,\sigma_j+k_j-1}p + f_{j,\sigma_j+k_j} \end{aligned}, \qquad j = 1, \cdots, m \tag{3.4.23e}$$

$$D_{iH}(p) = d_{i,0}p^{\sigma_i} + d_{i,1}p^{\sigma_i-1} + \cdots + d_{i,\sigma_i-k_{i+1}}p^{k_{i+1}}, \qquad i = 0, 1, \cdots, m \tag{3.4.23f}$$

The coefficients of $F_j(p)$, $j = 1, \cdots, m$, in Eq. (3.4.23e) are obtained from

33

$$
\begin{bmatrix} f_{j,0} \\ f_{j,1} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f_{j,\sigma_j+k_j} \end{bmatrix} = \begin{bmatrix} f_{j-1,\sigma_{j-1}-k_j+1} \\ f_{j-1,\sigma_{j-1}-k_j+2} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f_{j-1,\sigma_{j-1}+k_{j-1}} \end{bmatrix}
$$

$$
- \begin{bmatrix}
a_{j-1,\sigma_{j-1}-k_j+1} + \overline{a}_{\sigma_{j-1}-k_j+1,0} & a_{j-1,\sigma_{j-1}-k_j} + \overline{a}_{\sigma_{j-1}-k_j+1,1} & \cdots & a_{j-1,1} + \overline{a}_{\sigma_{j-1}-k_j+1,\sigma_{j-1}-k_j} \\
a_{j-1,\sigma_{j-1}-k_j+2} + \overline{a}_{\sigma_{j-1}-k_j+2,0} & a_{j-1,\sigma_{j-1}-k_j+1} + \overline{a}_{\sigma_{j-1}-k_j+2,1} & \ddots & a_{j-1,2} + \overline{a}_{\sigma_{j-1}-k_j+2,\sigma_{j-1}-k_j} \\
\vdots & \ddots & \ddots & \vdots \\
a_{j-1,k_{j-1}} + \overline{a}_{k_{j-1},0} & \ddots & \ddots & \vdots \\
\overline{a}_{k_{j-1}+1,0} & a_{j-1,k_{j-1}} + \overline{a}_{k_{j-1}+1,1} & \ddots & \vdots \\
\vdots & \overline{a}_{k_{j-1}+1,1} & \ddots & \vdots \\
\vdots & \ddots & \ddots & a_{j-1,k_{j-1}} + \overline{a}_{\sigma_{j-1}+k_{j-1}-k_j,\sigma_{j-1}-k_j} \\
\vdots & \ddots & \ddots & \overline{a}_{\sigma_{j-1}+k_{j-1}-k_j+1,\sigma_{j-1}-k_j} \\
\vdots & \ddots & \ddots & \vdots \\
\overline{a}_{\sigma_{j-1}+k_{j-1},0} & \overline{a}_{\sigma_{j-1}+k_{j-1},1} & \cdots & \overline{a}_{\sigma_{j-1}+k_{j-1},\sigma_{j-1}-k_j}
\end{bmatrix}
$$

$$
\times \begin{bmatrix} d_{j-1,0} \\ d_{j-1,1} \\ \vdots \\ d_{j-1,\sigma_{j-1}-k_j} \end{bmatrix}
$$

(3.4.24)

The coefficients of $D_{iH}(p)$, $i = 0, 1, \cdots, m$, in Eq. (3.4.23f) are obtained from

$$
\begin{bmatrix} d_{i,0} \\ d_{i,1} \\ \vdots \\ d_{i,\sigma_i-k_{i+1}-1} \\ d_{i,\sigma_i-k_{i+1}} \end{bmatrix} =
$$

$$
\begin{bmatrix}
a_{i,0} & 0 & \cdots & \cdots & 0 \\
a_{i,1}+\overline{a}_{1,0} & a_{i,0} & \ddots & \ddots & \vdots \\
\vdots & a_{i,1}+\overline{a}_{1,1} & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & 0 \\
a_{i,\sigma_i-k_{i+1}}+\overline{a}_{\sigma_i-k_{i+1},0} & a_{i,\sigma_i-k_{i+1}-1}+\overline{a}_{\sigma_i-k_{i+1},1} & \cdots & a_{i,1}+\overline{a}_{\sigma_i-k_{i+1},\sigma_i-k_{i+1}-1} & a_{i,0}
\end{bmatrix}^{-1}
$$

$$
\times \begin{bmatrix} f_{i,0} \\ f_{i,1} \\ \vdots \\ f_{i,\sigma_i-k_{i+1}-1} \\ f_{i,\sigma_i-k_{i+1}} \end{bmatrix}
$$

$$(3.4.25)$$

The degrees of the respective polynomials in Eq. (3.4.22) are given by

$$\deg\left[A_j(p)\right]=k_j \tag{3.4.26a}$$

$$\deg\left[B_j(p)\right]=k_{j+1}, \qquad\qquad k_{j+1}<k_j \tag{3.4.26b}$$

$$\deg\left[D_j(p)\right]\doteq\sigma_j=k_{j-1}-1 \tag{3.4.26c}$$

$$\deg\left[N_j(p)\right]=k_j-1 \tag{3.4.26d}$$

$$\deg\left[F_j(p)\right]=k_{j-1}+k_j-1 \tag{3.4.26e}$$

$$j=1,\cdots,m$$

The last Diophantine equation can be solved easily. With $i=m$,

$\deg\left[B_m(p)\right]=k_{m+1}=0$, we have

$$
\begin{bmatrix} d_{m,0} \\ \vdots \\ d_{m,\sigma_m} \\ n_{m,0} \end{bmatrix} = \begin{bmatrix} a_{m,0} & 0 & \cdots & 0 \\ a_{m,1}+\bar{a}_{1,0} & \ddots & \ddots & \vdots \\ \vdots & \ddots & a_{m,0} & 0 \\ \bar{a}_{\sigma_m+1,0} & \cdots & \cdots & b_{m,0} \end{bmatrix}^{-1} \begin{bmatrix} f_{m,0} \\ \vdots \\ f_{m,\sigma_m} \\ f_{m,\sigma_m+1} \end{bmatrix}
\tag{3.4.27}
$$

3.4.2 Backward Algorithm

Starting with a known pair, $D_m(p)$ and $N_m(p)$. The objective here is to determine the original pair, $D_0(p)$ and $N_0(p)$. From Eqs. (3.4.23d) and (3.4.23f), we have

$$
D_{m-1}(p) = D_{(m-1)H}(p) + N_m(p)
\tag{3.4.28a}
$$

The polynomial $N_{m-1}(p)$ can be solved from Eqs. (3.4.23c) and (3.4.23d) as

$$
N_{m-1}(p) = D_m(p) - N_m(p)Q_{m-1}(p)
\tag{3.4.28b}
$$

Continuing on in this way, with the help of Eqs. (3.4.23c), (3.4.23d) and (3.4.23f), we have the following backward recursive formulas:

$$
D_{i-1}(p) = D_{(i-1)H}(p) + N_i(p), \qquad\qquad i = 1, \cdots, m \tag{3.4.29a}
$$

$$
\begin{aligned}
N_{i-1}(p) &= D_i(p) - N_i(p)Q_{i-1}(p) \\
&= n_{i-1,0}p^{k_{i-1}-1} + n_{i-1,1}p^{k_{i-1}-2} + \cdots + n_{i-1,k_{i-1}-2}p + n_{i-1,k_{i-1}-1}
\end{aligned}, \qquad i = 1, \cdots, m \tag{3.4.29b}
$$

The coefficients of $N_{i-1}(p)$ in Eq. (3.4.29b) are given by

36

$$
\begin{bmatrix} n_{i-1,0} \\ n_{i-1,1} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ n_{i-1,k_{i-1}-1} \end{bmatrix}
=
\begin{bmatrix} d_{i,0} \\ d_{i,1} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ d_{i,k_{i-1}-1} \end{bmatrix}
$$

$$
-
\begin{bmatrix}
q_{i-1,0} & 0 & \cdots & 0 \\
q_{i-1,1}+\overline{q}_{1,0} & q_{i-1,0} & \ddots & \vdots \\
q_{i-1,2}+\overline{q}_{2,0} & q_{i-1,1}+\overline{q}_{2,1} & \ddots & 0 \\
\vdots & \ddots & \ddots & q_{i-1,0} \\
q_{i-1,k_{i-1}-k_i}+\overline{q}_{k_{i-1}-k_i,0} & q_{i-1,k_{i-1}-k_i-1}+\overline{q}_{k_{i-1}-k_i,1} & \ddots & q_{i-1,1} \\
\overline{q}_{k_{i-1}-k_i+1,0} & q_{i-1,k_{i-1}-k_i}+\overline{q}_{k_{i-1}-k_i+1,1} & \ddots & q_{i-1,2} \\
\vdots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \vdots \\
\overline{q}_{k_{i-1},0} & \overline{q}_{k_{i-1},1} & \cdots & q_{i-1,k_{i-1}-k_i}
\end{bmatrix}
\begin{bmatrix} n_{i,0} \\ n_{i,1} \\ \vdots \\ n_{i,k_{i-1}} \end{bmatrix}
\tag{3.4.30}
$$

where q's are the coefficients of $Q_{i-1}(p)$, i.e.

$$
Q_{i-1}(p) = q_{i-1,0}\,p^{k_{i-1}-k_i} + q_{i-1,1}\,p^{k_{i-1}-k_i-1} + \cdots + q_{i-1,k_{i-1}-k_i-1}\,p + q_{i-1,k_{i-1}-k_i}
\tag{3.4.31}
$$

3.5 Example of Time-Varying Case

Consider a time-varying plant with input u and output y described as a second-order time-varying differential equation as

$$
[p^3 + 3p^2 + 4p + (1+e^{-t})]y = (p^2 + 2p + 1)u
$$

The associated Diophantine equation reads in this case

$$
D_0(p)A_0(p) + N_0(p)B_0(p) = F_0(p)
$$

37

where

$$A_0(p) = p^3 + 3p^2 + 4p + (1 + e^{-t})$$

$$B_0(p) = p^2 + 2p + 1$$

$$D_0(p) = d_{0,0}p^3 + d_{0,1}p^2 + d_{0,2}p + d_{0,3}$$

$$N_0(p) = n_{0,0}p^2 + n_{0,1}p + n_{0,2}$$

Placing the closed-loop poles at $s = (-1 \pm j)$, -1, and the observer poles at

$s = -1$, $s = -2$ and $s = -3$, respectively, leads to

$$\alpha(p) = (p + 1 + j)(p + 1 - j)(p + 1)$$

$$\delta(p) = (p + 1)(p + 2)(p + 3)$$

$$\begin{aligned}F_0(p) &= \delta(p)\alpha(p) \\ &= (p+1)(p+2)(p+3)(p+1+j)(p+1-j)(p+1) \\ &= p^6 + 9p^5 + 33p^4 + 65p^3 + 74p^2 + 46p + 12\end{aligned}$$

Appling Euclidean algorithm to $A_0(p)$ and $B_0(p)$ yields, according to Eq.

(3.3.2),

$$Q_0(p) = p + 1$$

$$R_0(p) = p + e^{-t}$$

$$Q_1(p) = p + (2 - e^{-t})$$

$$R_1(p) = 1 - e^{-t} + e^{-2t}$$

38

The Euclidean algorithm stops here since $\deg[R_1(p)] = 0$, i.e. $m = 1$ in Eq.

(3.3.2). Next we proceed to reduce the order of the Diophantine equation.

3.5.1 Forward Algorithm

Step 0:

Consider Eq. (3.4.25).

With $i = 0$, we can solve for the two leading coefficients of $D_0(p)$ as

$$\begin{bmatrix} d_{0,0} \\ d_{0,1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 9 \end{bmatrix} = \begin{bmatrix} 1 \\ 6 \end{bmatrix}$$

Then, from Eq. (3.4.23f), $D_{0H}(p)$ is given by

$$D_{0H}(p) = p^3 + 6p^2$$

From Eqs. (3.4.22) and (3.4.23), with $i = 1$, the following reduced order

Diophantine equation is obtained

$$D_1(p)A_1(p) + N_1(p)B_1(p) = F_1(p) \tag{3.5.1}$$

where

$$A_1(p) = p^2 + 2p + 1$$

$$B_1(p) = p + e^{-t}$$

$$F_1(p) = F_0(p) - D_{0H}(p)A_0(p) = f_{1,0}p^4 + f_{1,1}p^3 + f_{1,2}p^2 + f_{1,3}p + f_{1,4}$$

With the help of Eq. (3.4.13), the coefficients of $F_1(p)$ are given by

39

$$\begin{bmatrix} f_{1,0} \\ f_{1,1} \\ f_{1,2} \\ f_{1,3} \\ f_{1,4} \end{bmatrix} = \begin{bmatrix} 33 \\ 65 \\ 74 \\ 46 \\ 12 \end{bmatrix} - \begin{bmatrix} 4 & 3 \\ 1+e^{-t} & 4 \\ -3e^{-t} & 1+e^{-t} \\ 3e^{-t} & -2e^{-t} \\ -e^{-t} & e^{-t} \end{bmatrix} \begin{bmatrix} 1 \\ 6 \end{bmatrix}$$

$$= \begin{bmatrix} 11 \\ 40-e^{-t} \\ 68-3e^{-t} \\ 49+9e^{-t} \\ 12-5e^{-t} \end{bmatrix}$$

Step 1:

Repeating the above procedure and from Eq. (3.4.23), we solve for the two leading coefficients in $D_1(p)$ as

$$\begin{bmatrix} d_{1,0} \\ d_{1,1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 11 \\ 40-e^{-t} \end{bmatrix} = \begin{bmatrix} 11 \\ 18-e^{-t} \end{bmatrix}$$

So that $D_1(p)$ will be split into an "known part and an "unknown" part as

$$D_{1H}(p) = 11p^2 + (18-e^{-t})p$$

$$D_{1L}(p) = d_{1,2}$$

From Eqs. (3.4.22) and (3.4.23), with $i = 2$, the following reduced order Diophantine equation is obtained

$$D_2(p)A_2(p) + N_2(p)B_2(p) = F_2(p) \tag{3.5.2}$$

Where

$$A_2(p) = p + e^{-t}$$

$$B_2(p) = 1 - e^{-t} + e^{-2t}$$

$$F_2(p) = F_1(p) - D_{1H}(p)A_1(p)$$
$$= (22 - e^{-t})p^2 + (28 + 10e^{-t})p + (12 - 5e^{-t})$$

Step 2:

Repeating the above procedure and from Eq. (3.4.23), we solve for the two leading coefficients in $D_2(p)$ as

$$\begin{bmatrix} d_{2,0} \\ d_{2,1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ e^{-t} & 1 \end{bmatrix}^{-1} \begin{bmatrix} 22 - e^{-t} \\ 28 + 10e^{-t} \end{bmatrix} = \begin{bmatrix} 21 - e^{-t} \\ 28 - 11e^{-t} + e^{-2t} \end{bmatrix}$$

Notice that, in this final step, $D_{2L}(p) = 0$ and

$$D_2(p) = D_{2H}(p)$$
$$= (21 - e^{-t})p + (28 - 11e^{-t} + e^{-2t})$$

With known $D_2(p)$, $N_2(p)$ follows from the Diophantine equation (3.5.2) as

$$N_2(p) = n_{2,0} = \frac{12 - 12e^{-t} + 10e^{-2t} - e^{-3t}}{1 - e^{-t} + e^{-2t}}$$

3.5.2 Backward Algorithm

With known $D_2(p)$ and $N_2(p)$, $D_1(p)$ and $N_1(p)$ can be obtained from Eq. (3.4.29) with $i = 2$ as

$$D_1(p) = D_{1H}(p) + N_2(p)$$
$$= 11p^2 + (18 - e^{-t})p + \frac{12 - 12e^{-t} + 10e^{-2t} - e^{-3t}}{1 - e^{-t} + e^{-2t}}$$

$$N_1(p) = D_2(p) - N_2(p)Q_1(p)$$
$$= \frac{9 - 10e^{-t} + 12e^{-2t}}{1 - e^{-t} + e^{-2t}} p + \frac{4 - 3e^{-t} + 8e^{-2t}}{1 - e^{-t} + e^{-2t}}$$

Similarly, with known $D_1(p)$ and $N_1(p)$, $D_0(p)$ and $N_0(p)$ can be obtained

from Eq. (3.4.29) with $i = 1$ as

$$D_0(p) = D_{0H}(p) + N_1(p)$$
$$= p^3 + 6p^2 + \frac{9 - 10e^{-t} + 12e^{-2t}}{1 - e^{-t} + e^{-2t}} p + \frac{4 - 3e^{-t} + 8e^{-2t}}{1 - e^{-t} + e^{-2t}}$$

$$N_0(p) = D_1(p) - N_1(p)Q_0(p)$$
$$= \frac{2 - e^{-t} - e^{-2t}}{1 - e^{-t} + e^{-2t}} p^2 + \frac{5 - 6e^{-t} + e^{-2t} - e^{-3t}}{1 - e^{-t} + e^{-2t}} p + \frac{8 - 9e^{-t} + 2e^{-2t} - e^{-3t}}{1 - e^{-t} + e^{-2t}}$$

which is the solution of our original Diophantine equation.

3.6 Problem of Zero-Crossing

Unlike time-invariant case, time-varying systems will have zero-crossing

problem. Here is an example to demonstrate this problem, and also propose a method to

address this issue. Consider a time-varying plant with input u and output y described as

a second-order time-varying differential equation as

$$[p^2 + (20 + 12\sin 2t)p + 30\cos 2t]y = (p + 1)u$$

The associated Diophantine equation reads in this case

$$D_0(p)A_0(p) + N_0(p)B_0(p) = F_0(p)$$

where

$$A_0(p) = p^2 + (20 + 12\sin 2t)p + 30\cos 2t$$

$$B_0(p) = p + 1$$

$$D_0(p) = d_{0,0}p^2 + d_{0,1}p + d_{0,2}$$

$$N_0(p) = n_{0,0}p + n_{0,1}$$

Placing the closed-loop poles at $s = (-1 \pm j)$ and the observer poles at $s = -2$ and $s = -3$, respectively, leads to

$$\alpha(p) = (p + 1 + j)(p + 1 - j)$$

$$\delta(p) = (p + 2)(p + 3)$$

$$\begin{aligned}
F_0(p) &= \delta(p)\alpha(p) \\
&= (p + 1 + j)(p + 1 - j)(p + 2)(p + 3) \\
&= p^4 + 7p^3 + 18p^2 + 22p + 12
\end{aligned}$$

Appling Euclidean algorithm to $A_0(p)$ and $B_0(p)$ yields, according to Eq. (3.3.2),

$$Q_0(p) = p + (12\sin 2t + 19)$$

$$R_0(p) = 30\cos 2t - 12\sin 2t - 19$$

The Euclidean algorithm stops here since $\deg[R_0(p)] = 0$, i.e. $m = 0$ in Eq.

(3.3.2). Next we proceed to reduce the order of the Diophantine equation.

(a) Forward algorithm

Consider Eq. (3.4.25).

With $i = 0$, we can solve for the two leading coefficients of $D_0(p)$ as

$$\begin{bmatrix} d_{0,0} \\ d_{0,1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 20 + 12\sin(2t) & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 7 \end{bmatrix} = \begin{bmatrix} 1 \\ -13 - 12\sin(2t) \end{bmatrix}$$

Then, from Eq. (3.4.23f), $D_{0H}(p)$ is given by

$$D_{0H}(p) = p^2 + [-13 - 12\sin(2t)]p$$

From Eqs. (3.4.22) and (3.4.23), with $i = 1$, the following reduced order

Diophantine equation is obtained

$$D_1(p)A_1(p) + N_1(p)B_1(p) = F_1(p) \qquad (3.6.1)$$

where

$$A_1(p) = p + 1$$

$$B_1(p) = 30\cos 2t - 12\sin 2t - 19$$

$$F_1(p) = F_0(p) - D_{0H}(p)A_0(p) = f_{1,0}p^2 + f_{1,1}p + f_{1,2}$$

With the help of Eq. (3.4.13), the coefficients of $F_1(p)$ are given by

$$\begin{bmatrix} f_{1,0} \\ f_{1,1} \\ f_{1,2} \end{bmatrix} = \begin{bmatrix} 18 \\ 22 \\ 12 \end{bmatrix} - \begin{bmatrix} 78\cos(2t) & 20+12\sin(2t) \\ -168\sin(2t) & 54\cos(2t) \\ -120\cos(2t) & -60\sin(2t) \end{bmatrix} \begin{bmatrix} 1 \\ -13-12\sin(2t) \end{bmatrix}$$

$$= \begin{bmatrix} 278-78\cos(2t)+396\sin(2t)+144\sin(2t)^2 \\ 22+702\cos(2t)+168\sin(2t)+324\sin(4t) \\ 12[-29+10\cos(2t)+30\cos(4t)-65\sin(2t)] \end{bmatrix}$$

Repeating the above procedure and from Eq. (3.4.23), we solve for the two leading coefficients in $D_1(p)$ as

$$\begin{bmatrix} d_{1,0} \\ d_{1,1} \end{bmatrix} = \begin{bmatrix} 278-78\cos(2t)+396\sin(2t)+144\sin(2t)^2 \\ 4(-82+195\cos(2t)+18\cos(4t)-57\sin(2t)+81\sin(4t)) \end{bmatrix}$$

Notice that, in this final step, $D_{1L}(p)=0$ and

$$D_1(p) = D_{1H}(p) = [278-78\cos(2t)+396\sin(2t)+144\sin(2t)^2]p$$
$$+4(-82+195\cos(2t)+18\cos(4t)-57\sin(2t)+81\sin(4t)$$

With known $D_1(p)$, $N_1(p)$ follows from the Diophantine equation (3.6.1) as

$$N_1(p) = n_{1,0} = -\frac{4[5+165\cos(2t)-72\cos(4t)+138\sin(2t)+81\sin(4t)]}{-19+30\cos(2t)-12\sin(2t)}$$

(b) Backward algorithm

With known $D_1(p)$, $N_1(p)$, $D_0(p)$ and $N_0(p)$ can be obtained from Eq. (3.4.28) with $i=0$ as

$$D_0(p) = D_{0H}(p) + N_1(p) = d_{0,0}p^2 + d_{0,1}p + d_{0,2}$$

$$N_0(p) = D_1(p) - N_1(p)Q_0(p) = n_{0,0}p + n_{0,1}$$

where

$$d_{0,0} = 1$$

$$d_{0,1} = -13 - 12\sin(2t)$$

$$d_{0,2} = -\frac{4[5 + 165\cos(2t) - 72\cos(4t) + 138\sin(2t) + 81\sin(4t)]}{-19 + 30\cos(2t) - 12\sin(2t)}$$

$$n_{0,0} = 6[1927\cos(2t) + 381\cos(4t) - 2(848 + 90\cos(6t) + 967\sin(2t)$$
$$- 561\sin(4t) - 36\sin(6t))]/[-19 + 30\cos(2t) - 12\sin(2t)]$$

$$n_{0,1} = 12[1916 - 920\cos(2t) + 15\cos(4t) + 90\cos(6t) + 2168\sin(2t)$$
$$- 345\sin(4t) + 225\sin(6t)]/[-19 + 30\cos(2t) - 12\sin(2t)]$$

which is the solution of our original Diophantine equation.

Furthermore, the polynomials $n_u(p)$ and $n_y(p)$ in the controller structure can be obtained from Eqs. (3.1.15) and (3.4.1) as, respectively,

$$n_u(p) = D_0(p) - \delta(p) = \gamma_0 p + \gamma_1$$

$$n_y(p) = N_0(p) = \beta_0 p + \beta_1$$

where

$$\gamma_0 = -18 - 12\sin(2t)$$

$$\gamma_1 = \frac{94 - 840\cos(2t) + 288\cos(4t) - 480\sin(2t) - 324\sin(4t)]}{\Delta(t)}$$

$$\beta_0 = 6[1927\cos(2t) + 381\cos(4t) - 2(848 + 90\cos(6t) + 967\sin(2t)$$
$$- 561\sin(4t) - 36\sin(6t))]/\Delta(t)$$

$$\beta_1 = 12[1916 - 920\cos(2t) + 15\cos(4t) + 90\cos(6t) + 2168\sin(2t)$$
$$- 345\sin(4t) + 225\sin(6t)] / \Delta(t)$$

$$\Delta(t) = -19 + 30\cos(2t) - 12\sin(2t)$$

In order to avoid division by $\Delta(t)$, we can change the equations for the blocks

H_u and H_y in Figure 1, Eqs. (3.1.4) and (3.1.5), to

$$\frac{1}{\Delta(t)}\delta(p)B(p)v = n_u(p)B(p)u \qquad (3.6.2a)$$

$$\frac{1}{\Delta(t)}\delta(p)B(p)w = n_y(p)B(p)y \qquad (3.6.2b)$$

Since both $n_u(p)$ and $n_y(p)$ contain the factor $\dfrac{1}{\Delta(t)}$, it can be cancelled out on

both sides of the equations (3.6.2a) and (3.6.2b), and thus their implementations would

not create a problem.

With the change introduced in Eq. (3.6.2), it can easily be shown that the closed-

loop system equation becomes (instead of Eq. (3.1.13))

$$\delta(p)\alpha(p)y = \frac{1}{\Delta(t)}\delta(p)B(p)r \qquad (3.6.3)$$

To do away with the division by $\Delta(t)$, let the reference input be

$$r(t) = \Delta(t)r_1(t) + r_2(t) \qquad (3.6.4)$$

Substituting Eq. (3.6.4) into Eq. (3.6.3) yields

$$\delta(p)\alpha(p)y = \frac{1}{\Delta(t)}\delta(p)B(p)[\Delta(t)r_1(t) + r_2(t)]$$

$$= \delta(p)B(p)r_1(t) + [g(t) + \frac{1}{\Delta(t)}\delta(p)B(p)r_2(t)] \qquad (3.6.5)$$

where

$$g(t) = r_1(t)[\frac{1}{\Delta(t)}\delta(p)B(p)\mathbf{\Delta}(t)]$$

Selecting $r_2(t)$ in such a way that the bracketed term in Eq. (3.6.5) vanishes will

yield the final closed-loop system equation

$$\delta(p)\alpha(p)y(t) = \delta(p)B(p)r_1(t) \tag{3.6.6}$$

or

$$\alpha(p)y(t) = B(p)r_1(t) \tag{3.6.7}$$

$r_2(t)$ can be found from the following differential equation

$$\delta(p)B(p)r_2(t) = -\Delta(t)g(t)$$

or

$$\delta(p)B(p)r_2(t) = -r_1(t)[\delta(t)B(p)\mathbf{\Delta}(t)] \tag{3.6.8}$$

In our example, Eq. (3.6.8) reads

$$(p^3 + 6p^2 + 11p + 6)r_2(t) = -r_1(t)[-114 - 1668\cos(2t) + 18276\sin(2t)]$$

CHAPTER 4

SETTING UP CANONICAL FORMS FOR LINEAR TIME-VARYING SYSTEMS
FROM A DIFFERENTIAL EQUATION DESCRIPTION

In this chapter, a derivation of the observer and observability canonical forms starting from a single differential equation description is developed for linear time-varying systems. Initial condition conversions between the canonical forms and the differential equation description are given. With the help of the derived canonical forms, simulations of systems given by linear time-varying differential equations are greatly facilitated. General formulas for the canonical forms as well as their corresponding initial condition conversions are given.

4.1 Linear Time-Varying Systems Representations

Consider an nth order linear time-varying differential equation

$$
\begin{aligned}
&a_0(t)y^{(n)}(t)+a_1(t)y^{(n-1)}(t)+a_2(t)y^{(n-2)}(t)+\cdots+a_{n-1}(t)y^{(1)}(t)+a_n(t)y^{(0)}(t)\\
&=b_0(t)u^{(n)}(t)+b_1(t)u^{(n-1)}(t)+b_2(t)u^{(n-2)}(t)+\cdots+b_{n-1}(t)u^{(1)}(t)+b_n(t)u^{(0)}(t)
\end{aligned}
\tag{4.1.1}
$$

, $a_0(t)=1$

where $u(t)$ is the input, $y(t)$ is the output, the superscript (k), $k=0, \ldots , n,$ indicates the kth time-derivative, and $a_i(t)$ and $b_i(t)$, , $i=0, \ldots , n,$ are continuously differentiable functions of time. The coefficient a_0 is assumed to be unity without loss of generality. This system can be equivalently represented by a state variable description [29]. (The time argument t is dropped for brevity):

49

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$
(4.1.2)

where u and y are scalars, x is an n-vector, a dot represents differentiation with respect to time, and A, B, C, D are $n \times n$, $n \times 1$, $1 \times n$ and 1×1 matrices, respectively.

To obtain a state variable description of Eq. (4.1.2) from the differential description of Eq. (4.1.1), we propose here two different derivations. The first derivation is described in Section 4.2 which generates the "observability" canonical form. The second derivation described in Section 4.3 leads to the "observer" canonical form. Initial condition conversions for the two derived canonical forms are treated in Sections 4.4.

4.2 Observability Canonical Form of Linear Time-Varying Systems

The procedure of setting up the observability form is outlined below:

Rename Eq. (4.1.1) as

$$y^{(n)} + a_1 y^{(n-1)} + a_2 y^{(n-2)} + \cdots + a_{n-1} y^{(1)} + a_n y^{(0)}$$
$$= b_{0,0} u^{(n)} + b_{1,0} u^{(n-1)} + b_{2,0} u^{(n-2)} + \cdots + b_{n-1,0} u^{(1)} + b_{n,0} u^{(0)}$$
(4.2.1)

where

$$b_{i,0} = b_i, \qquad\qquad i = 0, 1, 2, \ldots, n.$$
(4.2.2)

The second subscript "0" in the coefficients b's is used to indicate the step number in the recursive process that follows.

Step 1 : Let the first state variable x_1 be defined as a linear combination of y and u, i.e.

$$x_1 = y - b_{0,0} u$$
(4.2.3a)

The output equation becomes

$$y = x_1 + b_{0,0}u \tag{4.2.3b}$$

Substituting y in Eq. (4.2.3b) into Eq. (4.2.1) yields

$$
\begin{aligned}
&x_1^{(n)} + a_1 x_1^{(n-1)} + a_2 x_1^{(n-2)} + \cdots + a_{n-1} x_1^{(1)} + a_n x_1 \\
&= b_{1,1} u^{(n-1)} + b_{2,1} u^{(n-2)} + \cdots + b_{n-1,1} u^{(1)} + b_{n,1} u
\end{aligned} \tag{4.2.4}
$$

where the coefficients of $u^{(i)}$ are given by

$$b_{n-i,1} = b_{n-i,0} - \sum_{k=0}^{n-i} \binom{k+i}{i} a_{n-i-k} b_{0,0}^{(k)}, \qquad i = 0,\ 1,\ 2,\ \ldots,\ n\text{-}1. \tag{4.2.5}$$

Notice that the order of the highest derivative of u in Eq. (4.2.4) is reduced to n-1.

Step 2 : Assigning the second state variable x_2 as

$$x_2 = x_1^{(1)} - b_{1,1}u \tag{4.2.6a}$$

leads to the state variable equation

$$x_1^{(1)} = x_2 + b_{1,1}u \tag{4.2.6b}$$

Substituting all the derivatives of x_1 in Eq. (4.2.4) with the help of Eq. (4.2.6b) gives rise to

$$
\begin{aligned}
&x_2^{(n-1)} + a_1 x_2^{(n-2)} + \cdots + a_{n-1} x_2 \\
&= -a_n x_1 + [b_{2,2} u^{(n-2)} + \cdots + b_{n-1,2} u^{(1)} + b_{n,2} u]
\end{aligned} \tag{4.2.7}
$$

where the coefficients of $u^{(i)}$ are given by

$$b_{n-i,2} = b_{n-i,1} - \sum_{k=0}^{n-1-i} \binom{k+i}{i} a_{n-1-i-k} b_{1,1}^{(k)}, \qquad i = 0,\ 1,\ 2,\ \ldots,\ n\text{-}2. \tag{4.2.8}$$

51

Notice that the order of the differential equation (4.2.7) is further reduced by one.

Step 3 : Assign the third state variable x_3 as

$$x_3 = x_2^{(1)} - b_{2,2}u \qquad (4.2.9a)$$

Then, the state variable equation for x_2 becomes

$$x_2^{(1)} = x_3 + b_{2,2}u \qquad (4.2.9b)$$

Substituting all the derivatives of x_2 in Eq. (4.2.7) with the help of Eq. (4.2.9b) results in

$$
\begin{aligned}
&x_3^{(n-2)} + a_1 x_3^{(n-3)} + \cdots + a_{n-2} x_3 \\
&= [-a_{n-1} x_2 - a_n x_1] + [b_{3,3} u^{(n-3)} + \cdots + b_{n-1,3} u^{(1)} + b_{n,3} u]
\end{aligned} \qquad (4.2.10)
$$

where the coefficient of $u^{(i)}$ are given by

$$b_{n-i,3} = b_{n-i,2} - \sum_{k=0}^{n-2-i} \binom{k+i}{i} a_{n-2-i-k} b_{2,2}^{(k)}, \qquad i = 0, 1, \ldots, n\text{-}3. \qquad (4.2.11)$$

Step 4 : Continue on with the above procedure until the *n*th state variable x_n is reached and is defined as

$$x_n = x_{n-1}^{(1)} - b_{n-1,n-1} u \qquad (4.2.12)$$

Then, the following equation is obtained (similar to Eqs. (4.2.7) and (4.2.10)):

$$
\begin{aligned}
&x_n^{(1)} + a_1 x_n \\
&= [-a_2 x_{n-1} - a_3 x_{n-2} - \cdots - a_{n-2} x_3 - a_{n-1} x_2 - a_n x_1] + b_{n,n} u
\end{aligned} \qquad (4.2.13)
$$

Step 5 : Collecting Eqs. (4.2.3b), (4.2.6b), (4.2.9b) on up to (4.2.13) yields the state variable description in the "observability" form.

52

$$
\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ -a_n & -a_{n-1} & \cdots & -a_2 & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} b_{1,1} \\ b_{2,2} \\ \vdots \\ b_{n-1,n-1} \\ b_{n,n} \end{bmatrix} u \qquad (4.2.14a)
$$

$$ y = x_1 + b_{0,0} u \qquad (4.2.14b) $$

where $\displaystyle b_{v,v} = b_{v,v-1} - \sum_{k=0}^{1} \binom{k+n-v}{n-v} a_{1-k} b_{v-1,v-1}{}^{(k)}$, $\qquad v = 1, 2, \ldots, n \qquad (4.2.15)$

in which the general formula for the coefficients b's are given by

$$ b_{n-i,j+1} = b_{n-i,j} - \sum_{k=0}^{n-j-i} \binom{k+i}{i} a_{n-j-i-k} b_{j,j}{}^{(k)} \qquad (4.2.16) $$

$j = 0, 1, \ldots , n\text{-}1$ and $i = 0, 1, \ldots , n\text{-}j\text{-}1$

The realization block diagram for Eq. (4.2.14) is shown in Figure 4.1, which is exactly the same structure as in the linear time invariant case.
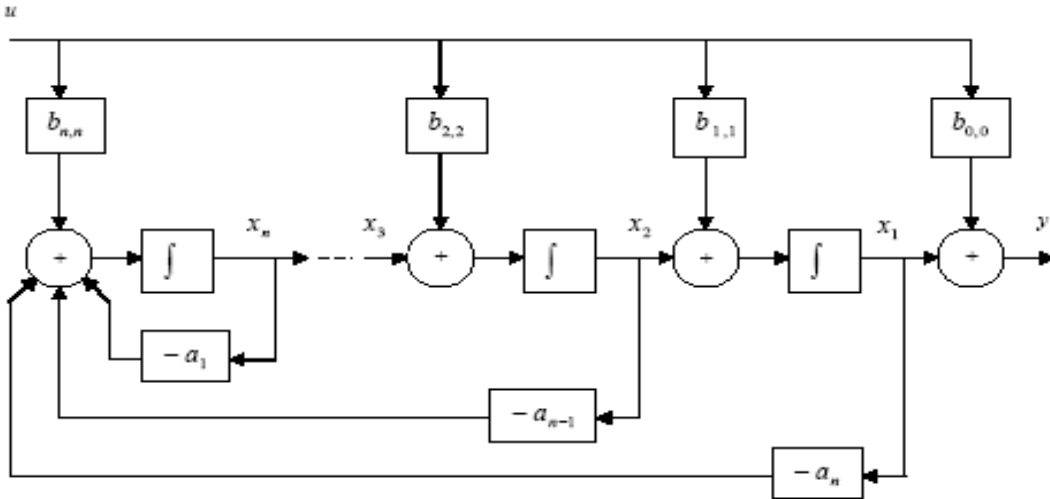

Figure 4.1. Observability canonical form

4.3 Observer Canonical Form of Linear Time-Varying Systems

The procedure of setting up the observer form is as follows:

Rename Eq. (4.1.1) as

$$y^{(n)} + a_{1,0}y^{(n-1)} + a_{2,0}y^{(n-2)} + \cdots + a_{n-1,0}y^{(1)} + a_{n,0}y^{(0)}$$
$$= b_{0,0}u^{(n)} + b_{1,0}u^{(n-1)} + b_{2,0}u^{(n-2)} + \cdots + b_{n-1,0}u^{(1)} + b_{n,0}u^{(0)} \tag{4.3.1}$$

where

$$a_{i,0} = a_i, \qquad\qquad i = 0, 1, 2, \ldots, n \tag{4.3.2a}$$

$$b_{i,0} = b_i, \qquad\qquad i = 0, 1, 2, \ldots, n. \tag{4.3.2b}$$

The second subscript "0" in the coefficients a's and b's is used to indicate the step number in the recursive process that follows.

Step 1 : Let the first state variable x_1 be defined as

$$x_1 = y - b_{0,0}u \tag{4.3.3a}$$

The output equation becomes

$$y = x_1 + b_{0,0}u \tag{4.3.3b}$$

Differentiating Eq. (4.3.3b) n times yields

$$y^{(n)} = x_1^{(n)} + \sum_{k=0}^{n}\binom{n}{k} b_{0,0}^{(n-k)} u^{(k)} \tag{4.3.4}$$

Substituting $y^{(n)}$ in Eq. (4.3.4) into Eq. (4.3.1) yields

$$x_1^{(n)} = -\sum_{k=0}^{n-1} a_{n-k,0}y^{(k)} - \sum_{k=0}^{n}\binom{n}{k} b_{0,0}^{(n-k)} u^{(k)} + \sum_{k=0}^{n} b_{n-k,0}u^{(k)}$$
$$= -\sum_{k=0}^{n-1} a_{n-k,1}y^{(k)} + \sum_{k=0}^{n-1} b_{n-k,1}u^{(k)} \tag{4.3.5}$$

where the coefficients of $y^{(k)}$ and $u^{(k)}$ are given by

$$a_{n-k,1} \doteq a_{n-k,0} - \binom{n}{k} a_{0,0}^{(n-k)}, \quad k = 0, 1, 2, \ldots, n\text{-}1 \tag{4.3.6a}$$

$$b_{n-k,1} \doteq b_{n-k,0} - \binom{n}{k} b_{0,0}^{(n-k)}, \quad k = 0, 1, 2, \ldots, n\text{-}1 \tag{4.3.6b}$$

Notice that the orders of the derivatives of y and u in Eq. (4.3.5) are reduced by one in comparison to Eq. (4.3.1), respectively. Rewrite Eq. (4.3.5) as

$$x_1^{(n)} = -a_{1,1} y^{(n-1)} + b_{1,1} u^{(n-1)} - \sum_{k=0}^{n-2} a_{n-k,1} y^{(k)} + \sum_{k=0}^{n-2} b_{n-k,1} u^{(k)} \tag{4.3.7}$$

Consider the first two R.H.S. terms of Eq. (4.3.7), $-a_{1,1} y^{(n-1)}$ and $b_{1,1} u^{(n-1)}$. They can be rewritten as

$$-a_{1,1} y^{(n-1)} = (-a_{1,1} y)^{(n-1)} + \sum_{k=0}^{n-2} \binom{n-1}{k} a_{1,1}^{(n-1-k)} y^{(k)} \tag{4.3.8a}$$

$$b_{1,1} u^{(n-1)} = (b_{1,1} u)^{(n-1)} - \sum_{k=0}^{n-2} \binom{n-1}{k} b_{1,1}^{(n-1-k)} u^{(k)} \tag{4.3.8b}$$

Eliminating $-a_{1,1} y^{(n-1)}$ and $b_{1,1} u^{(n-1)}$ from Eqs. (4.3.7) and (4.3.8) yields

$$x_1^{(n)} = (-a_{1,1} y)^{(n-1)} + (b_{1,1} u)^{(n-1)}$$
$$+ [-\sum_{k=0}^{n-2} a_{n-k,1} y^{(k)} + \sum_{k=0}^{n-2} b_{n-k,1} u^{(k)} + \sum_{k=0}^{n-2} \binom{n-1}{k} a_{1,1}^{(n-1-k)} y^{(k)} - \sum_{k=0}^{n-2} \binom{n-1}{k} b_{1,1}^{(n-1-k)} u^{(k)}]$$
$$\tag{4.3.9}$$

Let the square bracketed term on the R.H.S. of Eq. (4.3.9) be defined as the (n-1)th derivative of the second state variable x_2, i.e.

$$x_2^{(n-1)} = -\sum_{k=0}^{n-2}[a_{n-k,1} - \binom{n-1}{k}a_{1,1}^{(n-1-k)}]y^{(k)} + \sum_{k=0}^{n-2}[b_{n-k,1} - \binom{n-1}{k}b_{1,1}^{(n-1-k)}]u^{(k)}$$

$$= -\sum_{k=0}^{n-2}[a_{n-k,2}]y^{(k)} + \sum_{k=0}^{n-2}[b_{n-k,2}]u^{(k)} \tag{4.3.10}$$

where the coefficients of $y^{(k)}$ and $u^{(k)}$ are given by

$$a_{n-k,2} \doteq a_{n-k,1} - \binom{n-1}{k}a_{1,1}^{(n-1-k)}, \qquad k = 0, 1, 2, \ldots, n\text{-}2 \tag{4.3.11a}$$

$$b_{n-k,2} \doteq b_{n-k,1} - \binom{n-1}{k}b_{1,1}^{(n-1-k)}, \qquad k = 0, 1, 2, \ldots, n\text{-}2 \tag{4.3.11b}$$

Notice that the orders of derivatives of y and u in Eq. (4.3.10) are further reduced by one, respectively.

Then, Eq. (4.3.9) becomes

$$x_1^{(n)} = (-a_{1,1}y)^{(n-1)} + (b_{1,1}u)^{(n-1)} + x_2^{(n-1)} \tag{4.3.12}$$

On integrating Eq. (4.3.12) n-1 times under zero initial conditions, we obtain the state differential equation for the first state variable

$$x_1^{(1)} = -a_{1,1}y + b_{1,1}u + x_2 \tag{4.3.13a}$$

Substituting for y from Eq. (4.3.3b) yields

$$x_1^{(1)} = -a_{1,1}x_1 + (-a_{1,1}b_{0,0} + b_{1,1})u + x_2 \tag{4.3.13b}$$

Step 2 : Rewrite Eq. (4.3.10) as

$$x_2^{(n-1)} = -a_{2,2}y^{(n-2)} + b_{2,2}u^{(n-2)} - \sum_{k=0}^{n-3}[a_{n-k,2}]y^{(k)} + \sum_{k=0}^{n-3}[b_{n-k,2}]u^{(k)} \tag{4.3.14}$$

Consider the first two R.H.S. terms of Eq. (4.3.14), $-a_{2,2}y^{(n-2)}$ and $b_{2,2}u^{(n-2)}$.

They can be rewritten as

$$-a_{2,2}y^{(n-2)} = (-a_{2,2}y)^{(n-2)} + \sum_{k=0}^{n-3}\binom{n-2}{k}a_{2,2}^{(n-2-k)}y^{(k)} \tag{4.3.15a}$$

$$b_{2,2}u^{(n-2)} = (b_{2,2}u)^{(n-2)} - \sum_{k=0}^{n-3}\binom{n-2}{k}b_{2,2}^{(n-2-k)}u^{(k)} \tag{4.3.15b}$$

Eliminating $-a_{2,2}y^{(n-2)}$ and $b_{2,2}u^{(n-2)}$ from Eqs. (4.3.14) and (4.3.15) yields

$$x_2^{(n-1)} = (-a_{2,2}y)^{(n-2)} + (b_{2,2}u)^{(n-2)}$$
$$+ [-\sum_{k=0}^{n-3}a_{n-k,2}y^{(k)} + \sum_{k=0}^{n-3}b_{n-k,2}u^{(k)} + \sum_{k=0}^{n-3}\binom{n-2}{k}a_{2,2}^{(n-2-k)}y^{(k)} - \sum_{k=0}^{n-3}\binom{n-2}{k}b_{2,2}^{(n-2-k)}u^{(k)}]$$
$$\tag{4.3.16}$$

Let the square bracketed term from the R.H.S. of Eq. (4.3.16) be defined as the

(n-2)th derivative of the third state variable x_3, i.e.

$$x_3^{(n-2)} = -\sum_{k=0}^{n-3}[a_{n-k,2} - \binom{n-2}{k}a_{2,2}^{(n-2-k)}]y^{(k)} + \sum_{k=0}^{n-3}[b_{n-k,2} - \binom{n-2}{k}b_{2,2}^{(n-2-k)}]u^{(k)}$$
$$= -\sum_{k=0}^{n-3}[a_{n-k,3}]y^{(k)} + \sum_{k=0}^{n-3}[b_{n-k,3}]u^{(k)}$$

$$\tag{4.3.17}$$

where the coefficients of $y^{(k)}$ and $u^{(k)}$ are given by

$$a_{n-k,3} \doteq a_{n-k,2} - \binom{n-2}{k}a_{2,2}^{(n-2-k)}, \qquad k = 0, 1, 2, ..., n\text{-}3 \tag{4.3.18a}$$

$$b_{n-k,3} \doteq b_{n-k,2} - \binom{n-2}{k}b_{2,2}^{(n-2-k)}, \qquad k = 0, 1, 2, ..., n\text{-}3 \tag{4.3.18b}$$

Again, the order of derivatives of y and u in Eq. (4.3.17) are further reduced by

one, respectively.

Eq. (4.3.16) becomes

$$x_2^{(n-1)} = (-a_{2,2}y)^{(n-2)} + (b_{2,2}u)^{(n-2)} + x_3^{(n-2)} \tag{4.3.19}$$

Integrating Eq. (4.3.19) n-2 times leads to

$$x_2^{(1)} = -a_{2,2}y + b_{2,2}u + x_3 \tag{4.3.20a}$$

Utilizing Eq. (4.3.3b) yields the second state differential equation:

$$x_2^{(1)} = -a_{2,2}x_1 + (-a_{2,2}b_{0,0} + b_{2,2})u + x_3 \tag{4.3.20b}$$

Step 3 : Continue on with the same process as in Steps 1 and 2 until the derivative of the (n-1)th state variable x_{n-1} is reached (in a pattern similar to Eqs. (4.3.12) and (4.3.19))

$$x_{n-1}^{(2)} = (-a_{n-1,n-1}y)^{(1)} + (b_{n-1,n-1}u)^{(1)} + x_n^{(1)} \tag{4.3.21}$$

Integrating Eq. (4.3.21) and utilizing Eq. (4.3.3b) results in

$$x_{n-1}^{(1)} = -a_{n-1,n-1}x_1 + (-a_{n-1,n-1}b_{0,0} + b_{n-1,n-1})u + x_n \tag{4.3.22}$$

Then, the last state variable equation is obtained as

$$x_n^{(1)} = -a_{n,n}x_1 + (-a_{n,n}b_{0,0} + b_{n,n})u \tag{4.3.23}$$

Step 4 : Collecting Eqs. (4.3.3b), (4.3.13b), (4.3.20b) on up to (4.3.23) yields the sought-after state variable description in the "observer" form.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} -a_{1,1} & 1 & 0 & \cdots & 0 \\ -a_{2,2} & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & 0 & \ddots & 0 \\ -a_{n-1,n-1} & 0 & \ddots & \ddots & 1 \\ -a_{n,n} & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} -a_{1,1}b_{0,0} + b_{1,1} \\ -a_{2,2}b_{0,0} + b_{2,2} \\ \vdots \\ -a_{n-1,n-1}b_{0,0} + b_{n-1,n-1} \\ -a_{n,n}b_{0,0} + b_{n,n} \end{bmatrix} u \tag{4.3.24a}$$

$$y = x_1 + b_{0,0}u \tag{4.3.24b}$$

where

$$a_{v,v} = a_{v,v-1} - \binom{n-v+1}{n-v} a_{v-1,v-1}^{(1)}, \qquad v = 1, 2, \ldots, n \tag{4.3.25a}$$

$$b_{v,v} = b_{v,v-1} - \binom{n-v+1}{n-v} b_{v-1,v-1}^{(1)}, \qquad v = 1, 2, \ldots, n \tag{4.3.25b}$$

in which the general formulas for the coefficients a's and b's are given by

$$a_{n-k,j+1} = a_{n-k,j} - \binom{n-j}{k} a_{j,j}^{(n-j-k)} \tag{4.3.26a}$$

$$b_{n-k,j+1} = b_{n-k,j} - \binom{n-j}{k} b_{j,j}^{(n-j-k)} \tag{4.3.26b}$$

$j = 0, 1, \ldots, n\text{-}1$ and $k = 0, 1, \ldots, n\text{-}j\text{-}1$

The realization block diagram of Eq. (4.3.24) is shown in Figure 4.2. , which is exactly the same structure as in the linear time invariant case.
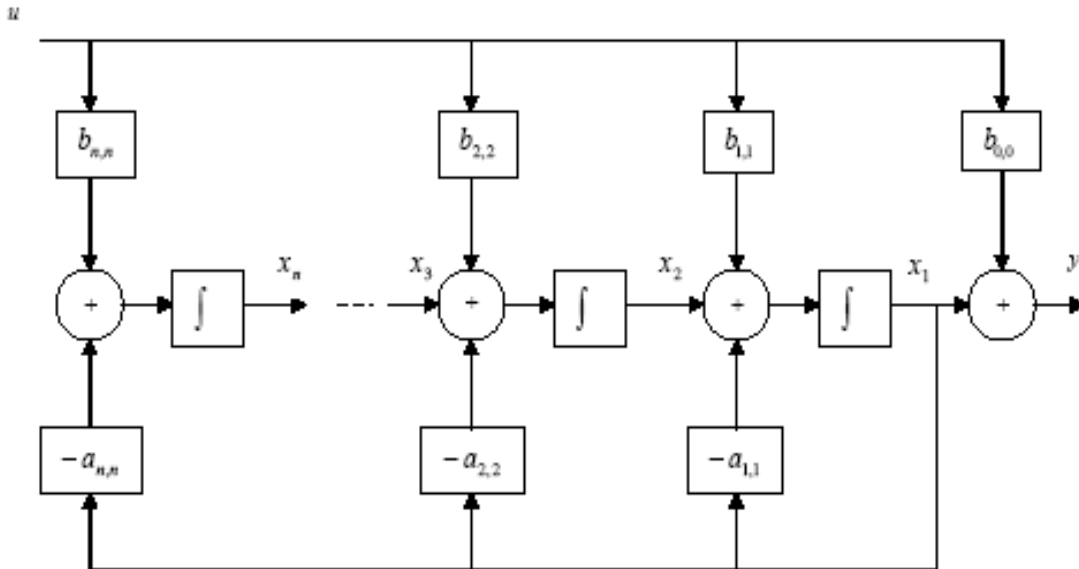
Figure 4.2. Observer canonical form

<center>4.4 Initial Condition Conversion</center>

4.4.1 Initial condition conversion for observability canonical form

The objective here is to set up the equation

$$\mathbf{y} = \mathbf{O}\mathbf{x} + \mathbf{T}\mathbf{u} \tag{4.4.1}$$

where

$$\mathbf{y} = [y \quad y^{(1)} \quad \cdots \quad y^{(n-1)}]^T \tag{4.4.2a}$$

$$\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_n]^T \tag{4.4.2b}$$

$$\mathbf{u} = [u \quad u^{(1)} \quad \cdots \quad u^{(n-1)}]^T \tag{4.4.2c}$$

The matrices $\mathbf{O}$ and $\mathbf{T}$ are both $n \times n$. In the time-invariant case, $\mathbf{O}$ is the observability matrix and $\mathbf{T}$ is the Toeplitz matrix [1]. To begin with, we differentiate Eq. (4.2.14b) repeatedly to obtain, with the help of Eq. (4.2.14a),

$$y^{(1)} = x_1^{(1)} + [b_{0,0}u]^{(1)} = x_2 + b_{1,1}u + [b_{0,0}u]^{(1)} \tag{4.4.3a}$$

$$y^{(2)} = x_2^{(1)} + [b_{1,1}u]^{(1)} + [b_{0,0}u]^{(2)} = x_3 + b_{2,2}u + [b_{1,1}u]^{(1)} + [b_{0,0}u]^{(2)} \tag{4.4.3b}$$

$$y^{(3)} = x_3^{(1)} + [b_{2,2}u]^{(1)} + [b_{1,1}u]^{(2)} + [b_{0,0}u]^{(3)}$$
$$= x_4 + b_{3,3}u + [b_{2,2}u]^{(1)} + [b_{1,1}u]^{(2)} + [b_{0,0}u]^{(3)} \tag{4.4.3c}$$

$$\vdots$$

$$y^{(i-1)} = x_i + \sum_{k=0}^{i-1}[b_{i-1-k,i-1-k}u]^{(k)} \tag{4.4.3d}$$

$$\vdots$$

$$y^{(n-1)} = x_n + \sum_{k=0}^{n-1}[b_{n-1-k,n-1-k}u]^{(k)} \tag{4.4.3e}$$

<center>60</center>

Using binomial expansion for the derivatives of a product, Eq. (4.4.3d) can be rewritten as

$$y^{(i-1)} = x_i + \sum_{k=0}^{i-1}\sum_{j=0}^{k}\binom{k}{j}b_{i-1-k,i-1-k}^{(j)}u^{(k-j)}, \quad i = 1, 2, \ldots, n \tag{4.4.4}$$

which can further be simplified to

$$y^{(i-1)} = x_i + \sum_{k=0}^{i-j}\binom{i-k-1}{i-j-k}b_{k,k}^{(i-j-k)}u^{(j-1)} \tag{4.4.5}$$

$i=1, 2, \ldots, n$ and $j=1, 2, \ldots, i$

Eq. (4.4.5) can be arranged in matrix form as

$$
\begin{bmatrix} y \\ y^{(1)} \\ \vdots \\ y^{(n-2)} \\ y^{(n-1)} \end{bmatrix} =
\begin{bmatrix}
o_{1,1} & o_{1,2} & \cdots & o_{1,n-1} & o_{1,n} \\
o_{2,1} & o_{2,2} & \ddots & \cdots & o_{2,n} \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
o_{n-1,1} & \vdots & \ddots & o_{n-1,n-1} & o_{n-1,n} \\
o_{n,1} & o_{n,2} & \cdots & o_{n,n-1} & o_{n,n}
\end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix}
$$

$$
+
\begin{bmatrix}
t_{1,1} & t_{1,2} & \cdots & t_{1,n-1} & t_{1,n} \\
t_{1,1} & t_{2,2} & \ddots & \cdots & t_{2,n} \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
t_{n-1,1} & \vdots & \ddots & t_{n-1,n-1} & t_{n-1,n} \\
t_{n,1} & t_{n,2} & \cdots & t_{n,n-1} & t_{n,n}
\end{bmatrix}
\begin{bmatrix} u \\ u^{(1)} \\ \vdots \\ u^{(n-2)} \\ u^{(n-1)} \end{bmatrix}
\tag{4.4.6}
$$

where

$$o_{i,j} = \begin{cases} 0 & , \quad i \neq j \\ 1 & , \quad i = j \end{cases} \tag{4.4.7a}$$

$$t_{i,j} = \begin{cases} 0 & , \quad i < j \\ \displaystyle\sum_{k=0}^{i-j}\binom{i-k-1}{i-j-k}b_{k,k}^{(i-j-k)} & , \quad i \geq j \end{cases} \tag{4.4.7b}$$

61

Setting $t=0$ and noting that $u(0) = u^{(1)}(0) = \cdots = u^{(n-1)}(0) = 0$ yields the initial condition conversion

$$\mathbf{y}(0) = \begin{bmatrix} y(0) \\ y^{(1)}(0) \\ \vdots \\ y^{(n-2)}(0) \\ y^{(n-1)}(0) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \\ \vdots \\ x_{n-1}(0) \\ x_n(0) \end{bmatrix} = \mathbf{O}(0)\mathbf{x}(0) \tag{4.4.8}$$

Remarks :

    (a)   Matrix $\mathbf{O}(0)$ is the identity matrix, which is the same as in the linear time-invariant case.

    (b)   For implementation, the initial conditions of $\mathbf{x}(0)$ can be injected at the outputs of the integrators as shown in Figure 4.3.
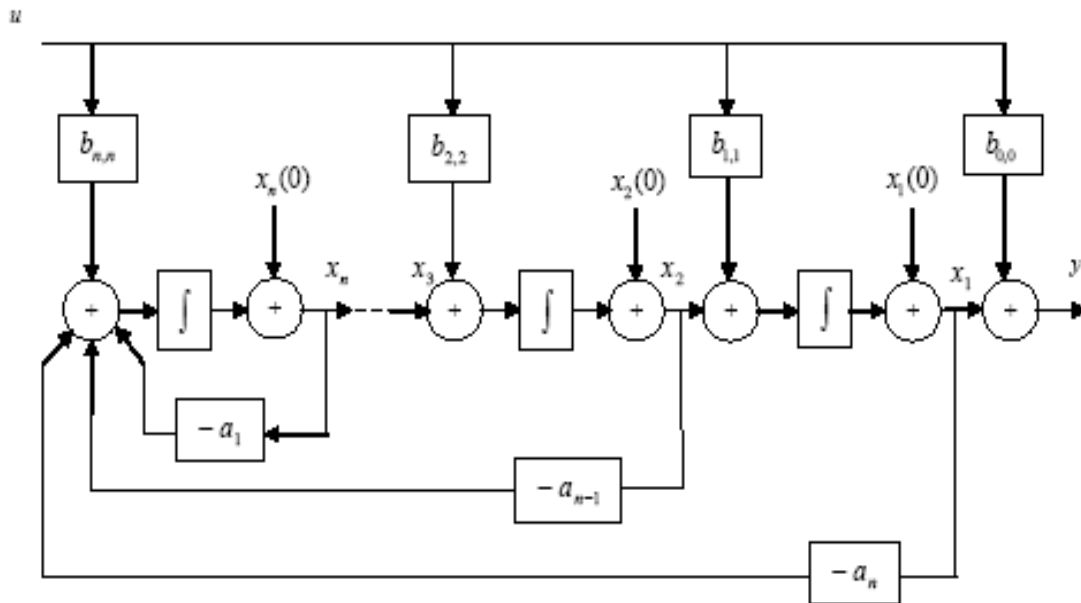
Figure 4.3. Observability canonical form with initial condition

4.4.2 Initial condition conversion for observer canonical form

Differentiating Eq. (4.3.24b) once gives

$$y^{(1)} = x_1^{(1)} + [b_{0,0}u]^{(1)} \qquad (4.4.9)$$

Substituting for $x_1^{(1)}$ from the state differential equation (4.3.24a) yields

$$y^{(1)} = -a_{1,1}x_1 + x_2 + [-a_{1,1}b_{0,0} + b_{0,0}^{(1)} + b_{1,1}]u + b_{0,0}u^{(1)} \qquad (4.4.10)$$

Differentiate Eq. (4.4.10) one more time and substituting for $x_2^{(1)}$ from Eq. (4.3.24a) gives

$$\begin{aligned}
y^{(2)} &= [-a_{1,1}x_1]^{(1)} + x_2^{(1)} + \{[-a_{1,1}b_{0,0} + b_{0,0}^{(1)} + b_{1,1}]u\}^{(1)} + [b_{0,0}u^{(1)}]^{(1)} \\
&= [a_{1,1}^{~2} - a_{1,1}^{(1)} - a_{2,2}]x_1 - a_{1,1}x_2 + x_3 \\
&+ [a_{1,1}^{~2}b_{0,0} - a_{1,1}b_{1,1} + a_{2,2}b_{0,0} + b_{2,2} + a_{1,1}^{(1)}b_{0,0}^{(1)} + b_{1,1}^{(1)} + b_{0,0}^{(2)}]u \\
&+ [-a_{1,1}b_{0,0} + b_{1,1} + 2b_{0,0}^{(1)}]u^{(1)} + b_{0,0}u^{(2)}
\end{aligned} \qquad (4.4.11)$$

Continue on in this fashion, the $(i\text{-}1)$th derivative, $y^{(i-1)}$, can be formed as

$$y^{(i-1)} = \sum_{j=1}^{i} o_{i,j}x_j + \sum_{j=1}^{i} t_{i,j}u^{(j-1)} \qquad (4.4.12)$$

$i=1, 2, \ldots, n$ and $j=1, 2, \ldots, i$

where

$$o_{i,j} = \begin{cases} 0 & , \quad i < j \\ 1 & , \quad i = j \\ \dot{o}_{i-1,1} - \sum_{k=1}^{i-1} o_{i-1,k}a_{k,k} & , \quad i = 2, 3, \cdots, n, \text{ and } j = 1 \\ \dot{o}_{i-1,j} + o_{i-1,j-1} & , \quad i = 2, 3, \cdots, n, \text{ and } j = 2, 3, \cdots, i-1 \end{cases} \qquad (4.4.13a)$$

$$t_{i,j} = \begin{cases} 0 & , \quad i < j \\ b_{0,0} & , \quad i = j \\ t_{i-1,1} + \sum_{k=1}^{i-1} o_{i-1,k}(-a_{k,k}b_{0,0} + b_{k,k}) & , \quad i = 2, 3, \cdots, n, \; and \; j = 1 \\ t_{i-1,j} + t_{i-1,j-1} & , \quad i = 2, 3, \cdots, n, \; and \; j = 2, 3, \cdots, i-1 \end{cases}$$

$$(4.4.13b)$$

and $a_{k,k}$ and $b_{k,k}$ are given by Eq. (4.3.25).

Setting $t=0$ and noting that $u(0) = u^{(1)}(0) = \cdots = u^{(n-1)}(0) = 0$ yields the initial condition conversion

$$\mathbf{y}(0) = \begin{bmatrix} y(0) \\ y^{(1)}(0) \\ \vdots \\ y^{(n-2)}(0) \\ y^{(n-1)}(0) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ o_{2,1}(0) & 1 & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ o_{n-1,1}(0) & \vdots & \ddots & 1 & 0 \\ o_{n,1}(0) & o_{n,2}(0) & \cdots & o_{n,n-1}(0) & 1 \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \\ \vdots \\ x_{n-1}(0) \\ x_n(0) \end{bmatrix} = \mathbf{O}(0)\mathbf{x}(0) \qquad (4.4.14)$$

where $o_{i,j}$ are defined in Eq. (4.4.13a).

Remarks :

(a) Notice that matrix $\mathbf{O}(0)$ is always invertible. It reduces to the well-known observability matrix of the "observer" form in the linear time-invariant case [1].

(b) The initial conditions of $\mathbf{x}(0)$ can be injected at the outputs of the integrators as shown in Figure 4.4.
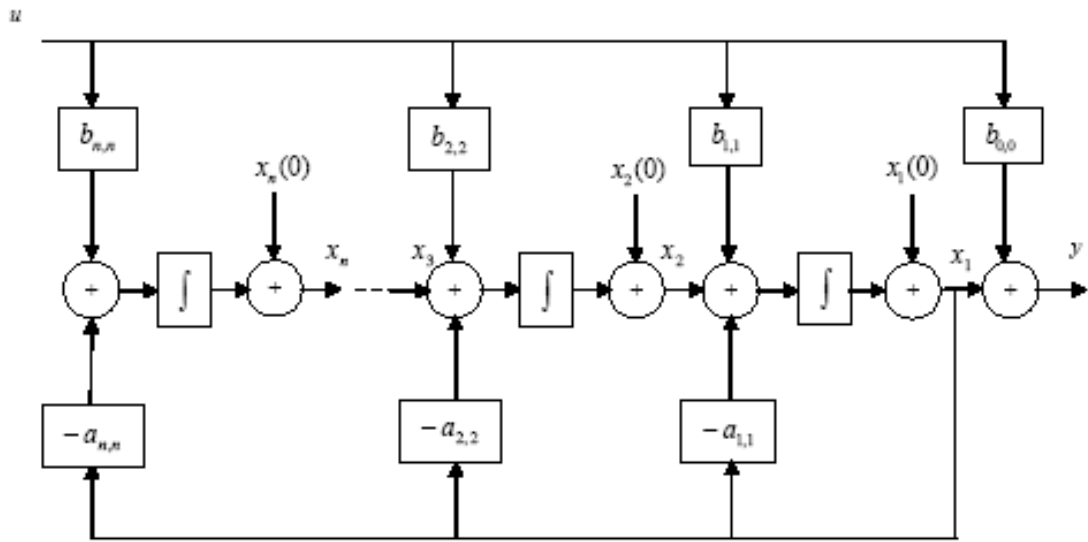
64

Figure 4.4. Observer canonical form with initial condition

CHAPTER 5

CASE STUDY

In this Chapter, the methods which were introduced before will be simulated in MATLAB - including Diophantine equation order reduction, pole-placement with combined observer-controller and two canonical forms. Since there is no general systematic approach for LTV systems, for the purpose of comparison, a second-order nonlinear system plant is selected for demonstration of simulations. In Section 5.1, the design procedures of pole-placement with combined observer-controller design will be demonstrated with the help of Diophantine equation order reduction and canonical forms of observability and observer. Then, the design procedures from existing well-known methods: input-state feedback linearization and input-output feedback linearization will be applied to the same plant in Section 5.2 and 5.3, respectively. All simulation results from Simulink in MATLAB are shown in Section 5.4.

5.1 Pole-Placement with Combined Observer-Controller

Consider a second-order nonlinear system plant (in p.213 in [35]) having state-variable equations:

$$\dot{x}_1 = -2x_1 + x_2 + \sin x_1 \tag{5.1.1a}$$

$$\dot{x}_2 = -x_2 \cos x_1 + u \cos(2x_1) \tag{5.1.1b}$$

For describing this plant in a single differential equation in Eq. (3.1.1), let

$$x_1 = y \tag{5.1.2a}$$

$$x_2 = \dot{y} + 2y - \sin y, \tag{5.1.2b}$$

and its corresponding differential equation can be derived as follows:

$$\ddot{y} + 2\dot{y} + (2\cos y - \frac{\sin y \cos y}{y})y = \cos(2y)u \tag{5.1.3}$$

Comparing with Eq. (4.1.1), all of the coefficients can be found as

$$a_0 = 1 \tag{5.1.4a}$$

$$a_1 = 2 \tag{5.1.4b}$$

$$a_2 = 2\cos y - \frac{\sin y \cos y}{y} \tag{5.1.4c}$$

$$b_0 = 0 \tag{5.1.4d}$$

$$b_1 = 0 \tag{5.1.4e}$$

$$b_2 = \cos(2y) \tag{5.1.4f}$$

Notice that the convention of subscripts in the coefficients a's and b's are different between Eq. (3.1.1) and Eq. (4.1.1). This is caused by the difference in degree requirements. The convention in Eq. (4.1.1) is adopted here for demonstrating the simulation.

Placing the closed-loop poles at $s = (-1 \pm j)$, and the observer poles at $s = -2$ and $s = -3$, respectively, leads $\alpha(p)$, $\delta(p)$ and $F(p)$ in Eq. (3.1.15c) to

$$\alpha(p) = (p+1+j)(p+1-j)$$

$$\delta(p) = (p+2)(p+3)$$

$$\begin{aligned}F(p) &= \delta(p)\alpha(p) \\ &= (p+2)(p+3)(p+1+j)(p+1-j) \\ &= p^4 + 7p^3 + 18p^2 + 22p + 12\end{aligned}$$

Referencing Eq. (3.4.5), the Diophantine equation can be written in matrix form as

$$\begin{bmatrix} a_0 & 0 & 0 & 0 & 0 \\ a_1 & a_0 & 0 & 0 & 0 \\ 2\dot{a}_1 + a_2 & a_1 & a_0 & 0 & 0 \\ \ddot{a}_1 + 2\dot{a}_2 & \dot{a}_1 + a_2 & a_1 & b_2 & 0 \\ \ddot{a}_2 & \dot{a}_2 & a_2 & \dot{b}_2 & b_2 \end{bmatrix} \begin{bmatrix} 1 \\ 5 + \gamma_0 \\ 6 + \gamma_1 \\ \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 7 \\ 18 \\ 22 \\ 12 \end{bmatrix}$$

Applying Diophantine equation order reduction, the unknowns γ_0, γ_1, β_0 and β_1 will be solved as

$$\gamma_0 = 0$$

$$\gamma_1 = 2 - a_2$$

$$\beta_0 = (10 - 2\dot{a}_2 - 5a_2 - 2\gamma_2)/b_2$$

$$\beta_1 = (12 - \ddot{a}_2 - 5\dot{a}_2 - 6a_2 - a_2\gamma_2 - \dot{b}_1\beta_1)/b_2$$

5.1.1 Design Procedure for Observability Canonical Form

There are three blocks: plant, H_u and H_y in Figure 3.1. Each block has its own input and output. The new state variables for each block can be generated by setting up Observability canonical form as described in Chapter 4.

5.1.1.1 Setting up Observability Canonical Form for the Plant

According to Eq. (3.1.1), the differential equation of the plant is

$$\ddot{y} + a_1\dot{y} + a_2 y = b_2 u$$

Let $y = x_{1\_plant} + b_0 u = x_{1\_plant}$

The state variables in observability canonical form associated to the plant can be found with the help of Eq. (4.2.14),

$$\dot{x}_{1\_plant} = x_{2\_plant} \tag{5.1.5a}$$

$$\dot{x}_{2\_plant} = -a_2 x_{1\_plant} - a_1 x_{2\_plant} + b_2 u \tag{5.1.5b}$$

When assigning $y(0) = 0.5$ and $\dot{y}(0) = 0.0$ the initial values of x_{1_plant} and x_{2_plant} in Eq. (5.1.5) can be found as

$$x_{1\_plant}(0) = y(0) = 0.5$$

$$x_{2\_plant}(0) = \dot{x}_{1\_plant}(0) = \dot{y}(0) = 0.0$$

5.1.1.2 Setting up Observability Canonical Form for H_u

From Eq. (3.1.4), the differential equation of H_u is

$$\dddot{v} + (\frac{2\dot{b}_2}{b_2} + 5)\ddot{v} + (\frac{\ddot{b}_2}{b_2} + \frac{5\dot{b}_2}{b_2} + 6)v = \gamma_2 u$$

To find the state variables in observability canonical form associated to H_u, first let $v = x_{1\_Hu} + b_0 u = x_{1\_Hu}$. Applying Eq. (4.2.14) yields

$$\dot{x}_{1\_Hu} = x_{2\_Hu}$$

$$\dot{x}_{2\_Hu} = -(\frac{\ddot{b}_2}{b_2} + \frac{5\dot{b}_2}{b_2} + 6)x_{1\_Hu} - (\frac{2\dot{b}_2}{b_2} + 5)x_{2\_Hu} + \gamma_2 u$$

5.1.1.3 Setting up Observability Canonical Form for H_y

From Eq. (3.1.5), the differential equation of H_y is

$$\dddot{w} + (\frac{2\dot{b}_2}{b_2} + 5)\ddot{w} + (\frac{\ddot{b}_2}{b_2} + \frac{5\dot{b}_2}{b_2} + 6)w = \beta_1 \dot{y} + (\frac{\beta_1 \dot{b}_2}{b_2} + \beta_2)y$$

To find the state variables in observability canonical form associated to H_y, let $w = x_{1\_Hy}$. Applying Eq. (4.2.14) leads to

$$\dot{x}_{1\_Hy} = x_{2\_Hy} + \beta_1 y$$

$$\dot{x}_{2\_Hy} = -(\frac{\ddot{b}_2}{b_2} + \frac{5\dot{b}_2}{b_2} + 6)x_{1\_Hy} - (\frac{2\dot{b}_2}{b_2} + 5)x_{2\_Hy} + [(\frac{\beta_1 \dot{b}_2}{b_2} + \beta_2) - \dot{\beta}_1 - (\frac{2\dot{b}_2}{b_2} + 5)\beta_1]y$$

5.1.1.4 Setting up Observability Canonical Form for Equivalent Overall System

From Eq. (3.1.14), the differential equation of the equivalent overall system is

$$\ddot{y} + 2\dot{y} + 2 = b_2 r$$

To find the state variables in observability canonical form associated to the equivalent overall system, let $y = x_{1\_equ}$. Applying Eq. (4.2.14) results in

$$\dot{x}_{1\_equ} = x_{2\_equ}$$

$$\dot{x}_{2\_equ} = -2x_{1\_equ} - 2x_{2\_equ} + b_2 \gamma$$

70

The initial values are

$$\dot{x}_{1\_equ}(0) = y(0) = 0.5$$

$$x_{2\_equ}(0) = \dot{x}_{1\_equ}(0) = \dot{y}(0) = 0$$

5.1.2 Design Procedure for Observer Canonical Form

The state variables for plant, H_u and H_y will be discussed in the following three sections. Also, the state variables in equivalent overall combined observer-controller is discussed in Section 5.1.2.4.

5.1.2.1 Setting up Observer Canonical Form for the Plant

According to Eq. (3.1.1), the differential equation of the plant is

$$\ddot{y} + a_1 \dot{y} + a_2 y = b_2 u$$

Let $y = x_{1\_plant} + b_0 u = x_{1\_plant}$

The state variables in observer canonical form associated to the plant can be found, with the help of Eq. (4.3.24), as

$$\dot{x}_{1\_plant} = -a_1 x_{1\_plant} + x_{2\_plant} \qquad (5.1.6a)$$

$$\dot{x}_{2\_plant} = -a_2 x_{1\_plant} + b_2 u \qquad (5.1.6b)$$

When assigning $y(0) = 0.5$ and $\dot{y}(0) = 0.0$ the initial values of x_{1_plant} in Eq. (5.1.6a) can be found as

$$x_{1\_plant}(0) = y(0) = 0.5$$

Knowing $\dot{y}(0) = \dot{x}_{1\_plant}(0) = -a_1 x_{1\_plant}(0) + x_{2\_plant}(0) = 0$ leads to

$$x_{2\_plant}(0) = 1.0$$

5.1.2.2 Setting up Observer Canonical Form for H_u

From Eq. (3.1.4), the differential equation of H_u is

$$\ddot{v} + (\frac{2\dot{b}_2}{b_2} + 5)\dot{v} + (\frac{\ddot{b}_2}{b_2} + \frac{5\dot{b}_2}{b_2} + 6)v = \gamma_2 u$$

To find the state variables in observer canonical form associated to H_u, first let

$v = x_{1\_Hu} + b_0 u = x_{1\_Hu}$. Applying Eq. (4.3.24) yields

$$\dot{x}_{1\_Hu} = -(\frac{2\dot{b}_2}{b_2} + 5)x_{1\_Hu} + x_{2\_Hu}$$

$$\dot{x}_{2\_Hu} = -[\frac{\ddot{b}_2}{b_2} + \frac{5\dot{b}_2}{b_2} + 6 - (\frac{2\dot{b}_2}{b_2} + 5)^{(1)}]x_{1\_Hu} + \gamma_2 u$$

5.1.2.3 Setting up Observer Canonical Form for H_y

From Eq. (3.1.5), the differential equation of H_y is

$$\ddot{w} + (\frac{2\dot{b}_2}{b_2} + 5)\dot{w} + (\frac{\ddot{b}_2}{b_2} + \frac{5\dot{b}_2}{b_2} + 6)w = \beta_1 \dot{y} + (\frac{\beta_1 \dot{b}_2}{b_2} + \beta_2)y$$

To find the state variables in observer canonical form associated to H_y, let

$w = x_{1\_Hy}$. Applying Eq. (4.3.24) leads to

$$\dot{x}_{1\_Hy} = -(\frac{2\dot{b}_2}{b_2} + 5)x_{1\_Hy} + x_{2\_Hy} + \beta_1 y$$

$$\dot{x}_{2\_Hy} = -[(\frac{\ddot{b}_2}{b_2} + \frac{5\dot{b}_2}{b_2} + 6) - (\frac{2\dot{b}_2}{b_2} + 5)^{(1)}]x_{1\_Hy} + (\frac{\beta_1 \dot{b}_2}{b_2} + \beta_2 - \dot{\beta}_1)y$$

5.1.2.4 Setting up Observer Canonical Form for Equivalent Overall System

From Eq. (3.1.14), the differential equation of the equivalent overall system is

72

$$\ddot{y} + 2\dot{y} + 2 = b_2 r$$

To find the state variables in observer canonical form associated to the equivalent overall system, let $y = x_{1\_equ}$. Applying Eq. (4.3.24) results in

$$\dot{x}_{1\_equ} = -a_1 x_{1\_equ} + x_{2\_equ} = -2x_{1\_equ} + x_{2\_equ}$$

$$\dot{x}_{2\_equ} = -2x_{1\_equ} + b_2 \gamma$$

The initial values are

$$\dot{x}_{1\_equ}(0) = y(0) = 0.5$$

$$x_{2\_equ}(0) = \dot{x}_{1\_equ}(0) + 2x_{1\_equ}(0) = 1.0$$

5.2 Input-State Feedback Linearization

Choosing new states from Eq. (5.1.1), with the method of input-state, produces

$$z_1 = x_1$$

$$z_2 = -2x_1 + x_2 + \sin(x_1)$$

Their derivatives are

$$\dot{z}_1 = \dot{x}_1 = z_2$$

$$\dot{z}_2 = -2z_2 - 2z_1 \cos(z_1) + \sin(z_1)\cos(z_1) + u\cos(2z_1)$$
$$= -2z_2 + v$$

respectively, where

$$v = -2z_1 \cos(z_1) + \sin(z_1)\cos(z_1) + u\cos(2z_1)$$

Computing the state transformation yields

73

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v$$

For the purpose of comparison, we can apply the same initial condition and place the poles as indicated in the Section 5.1.1.1 and 5.1.2.1. The poles are placed at $s = (-1 \pm j)$. The initial condition of x_1 and x_2 can be derived from Eqs. (5.1.2a) and (5.1.2b) with $t=0$ as

$$x_{1\_input-state}(0) = y(0) = 0.5$$

$$x_{2\_input-state}(0) = \dot{y}(0) + 2y(0) - \sin(y(0)) = 0.5205744614$$

5.3 Input-Output Feedback Linearization

Considering the state variables in Eq. (5.1.1), we have the choice of diffeomorphism such that

$$y = x_1 \tag{5.3.1a}$$

$$\dot{y} = \dot{x}_1 = -2x_1 + x_2 + \sin x_1 \tag{5.3.1b}$$

$$\begin{aligned} \ddot{y} = \ddot{x}_1 &= -2\dot{x}_1 - x_2 \cos(x_1) + u\cos(2x_1) + \dot{x}_1 \cos(x_1) \\ &= v \end{aligned} \tag{5.3.1c}$$

Using the concept of state-feedback linearization, we choose

$$u = \frac{1}{\cos(2x_1)}[v + 2\dot{x}_1 + x_2 \cos(x_1) - \dot{x}_1 \cos(x_1)]$$

$$v = -2\dot{y} - 2y \text{ when place poles at } s = (-1 \pm j)$$

Assigning the same initial condition in Sections 5.1.1.1 and 5.1.2.1, that is $y(0) = 0.5$ and $\dot{y}(0) = 0$. The initial condition of x_1 and x_2 can be found in Eqs. (5.3.1a) and (5.3.1b) with $t=0$ as

$$x_{1\_input-output}(0) = y(0) = 0.5$$

$$x_{2\_input-output}(0) = 0.5205744614$$

5.4 Simulations

Figure 5.1 to 5.20 show the signal waveforms of the simulation of the four methods. Zero input is applied to all simulation cases. The simulation case and their results for this example are summarized in table 5.1.

Table 5.1 Simulation cases for example in Section 5.1

Method	Initial condition	Simulation plots
Pole-placement using Observability form	$x_{1\_plant}(0) = y(0) = 0.5$ $x_{2\_plant}(0) = \dot{y}(0) = 0$	Figure 5.1
		Figure 5.2
		Figure 5.3
		Figure 5.4
		Figure 5.5
		Figure 5.6
Pole-placement using Observer form	$x_{1\_plant}(0) = y(0) = 0.5$ $\dot{y}(0) = 0$ $x_{2\_plant}(0) = 1.0$	Figure 5.7
		Figure 5.8
		Figure 5.9
		Figure 5.10
		Figure 5.11
		Figure 5.12
Input-state	$x_1(0) = y(0) = 0.5$ $x_2(0) = 0.5205744614$	Figure 5.16
Input-output		Figure 5.17

75

Figure 5.1 to 5.4 show the results from combined observer-controller in observability form including the states in all sub block diagrams. The output response from the equivalent overall system in observability form is plotted in Figure 5.5. Figure 5.6 shows the difference of output responses between observability form and the equivalent overall system. The difference is caused by the transient signal $\zeta(t)$ as indicated in Eq. (3.1.14).

Figure 5.7 to 5.10 illustrate the results from combined observer-controller in observer form including the states in all sub block diagrams. The output response of the equivalent overall system in observer form is shown in Figure 5.11. Figure 5.12 shows the difference of output responses between observer form and the equivalent overall system which is produced by the transient signal $\zeta(t)$ as indicated in Eq. (3.1.14). Figure 5.13 shows the output responses of observability and observer are identical. Figure 5.15 shows the difference of output responses from two equivalent overall systems. It is zero as expected.

The signal waveforms from input-state and input-output are plotted in Figure 5.16 and 5.17, respectively. The comparisons of combined observer-controller in two forms, input-state and input-output are shown in from Figure 5.18 to 5.20. The difference of output responses among equivalent overall system, input-state and output-state are zero in Figure 5.20 since they have same pole locations and initial conditions.
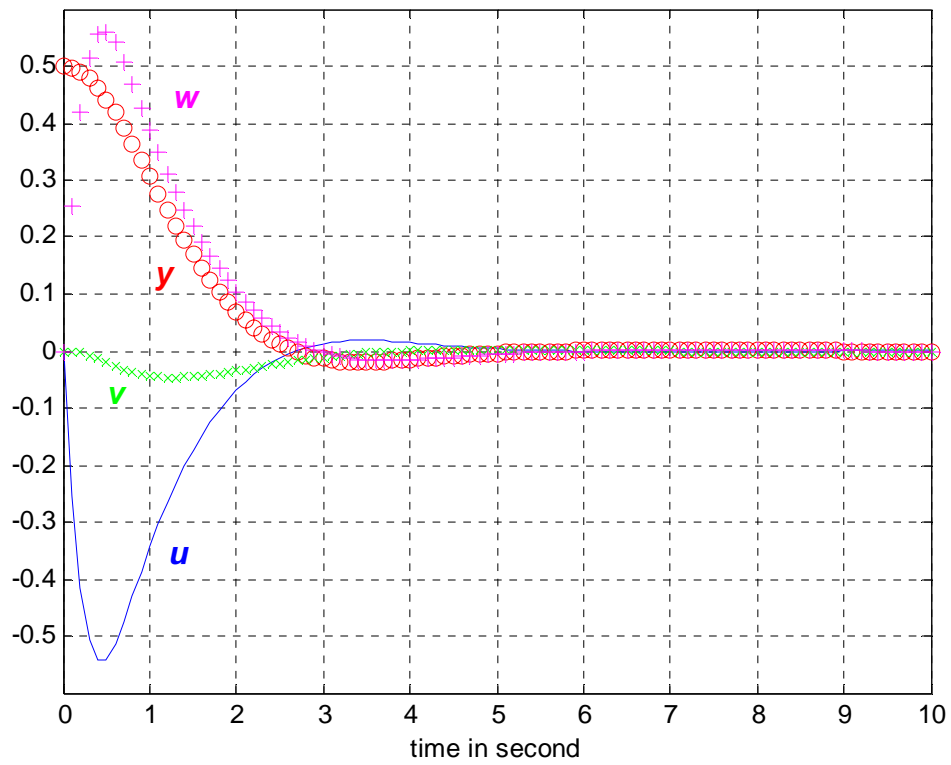
Figure 5.1. The zero input responses of the combined observer-controller in observability form with initial conditions: $y = 0.5$ and $\dot{y} = 0$
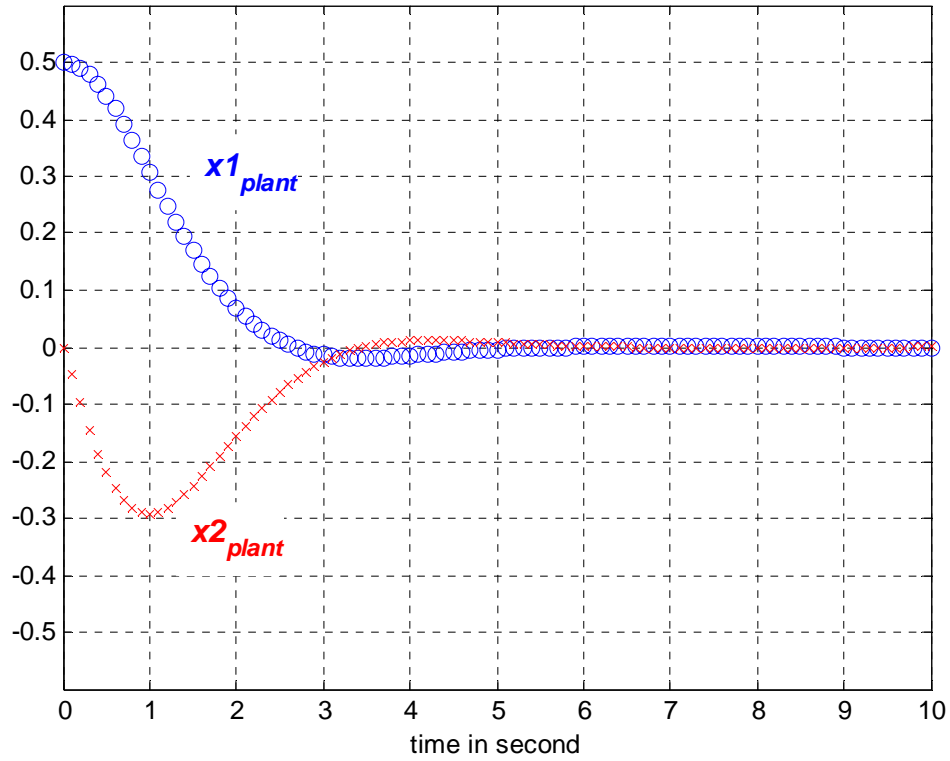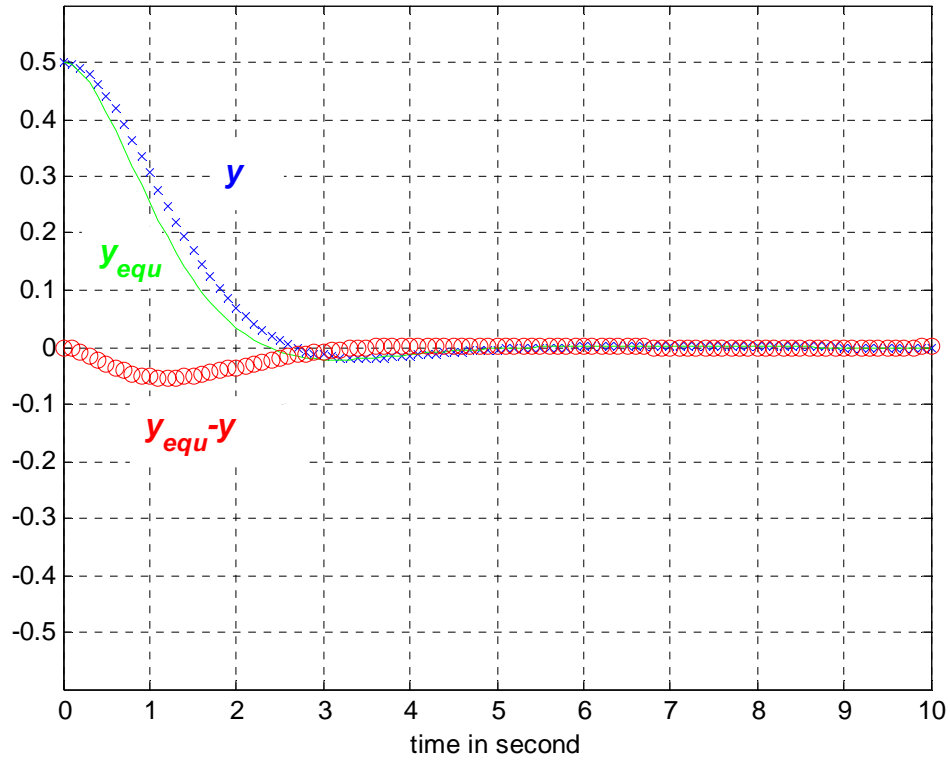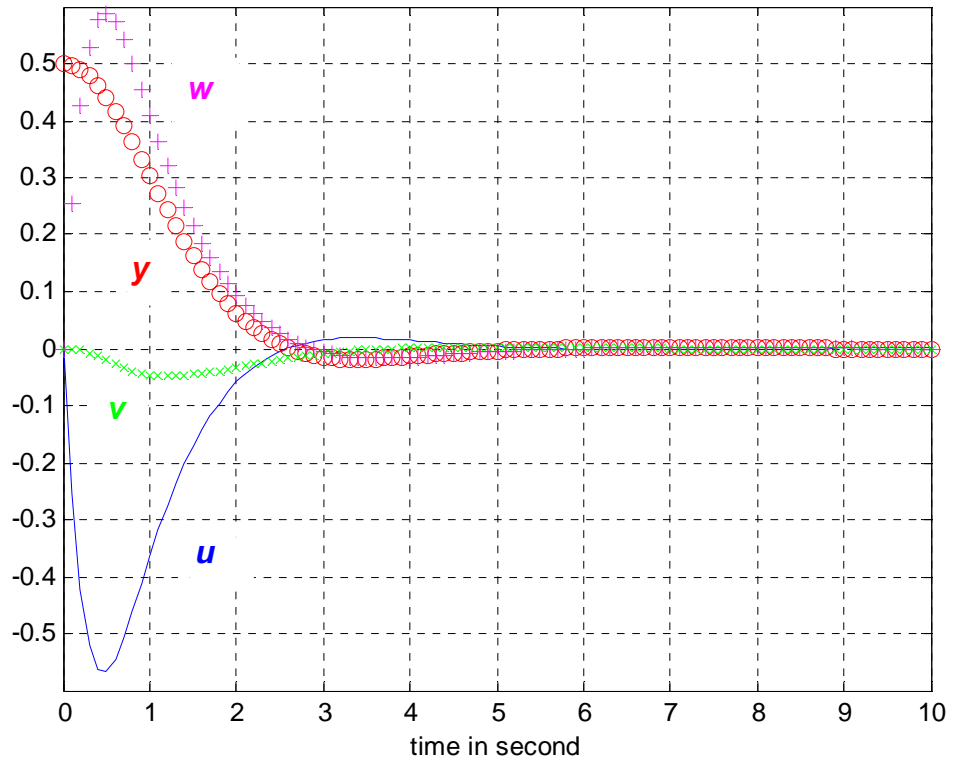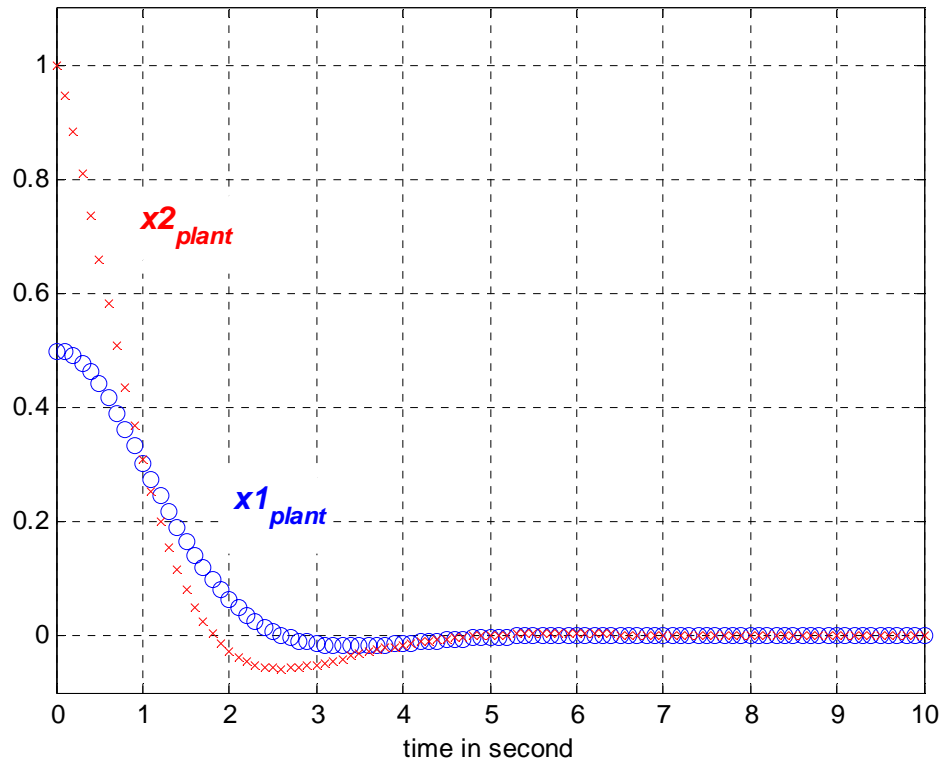
Figure 5.2. The zero input responses of the plant state variables in combined observer-controller in observability form with initial conditions: $x_{1\_plant}(0) = 0.5$ and
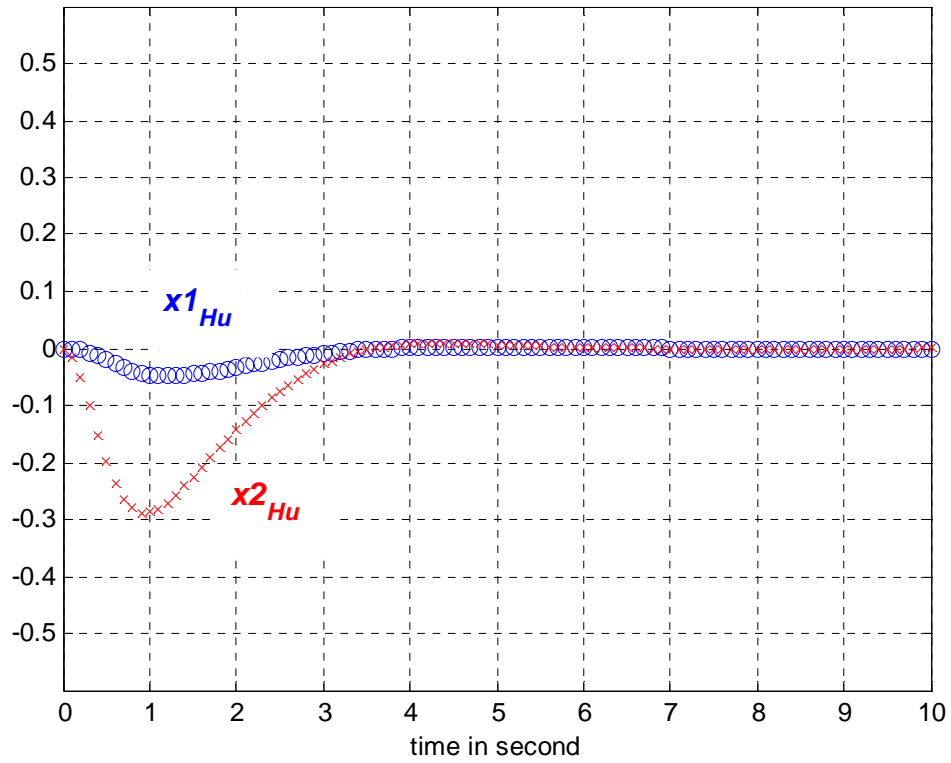
$$x_{2\_plant}(0) = 0.0$$
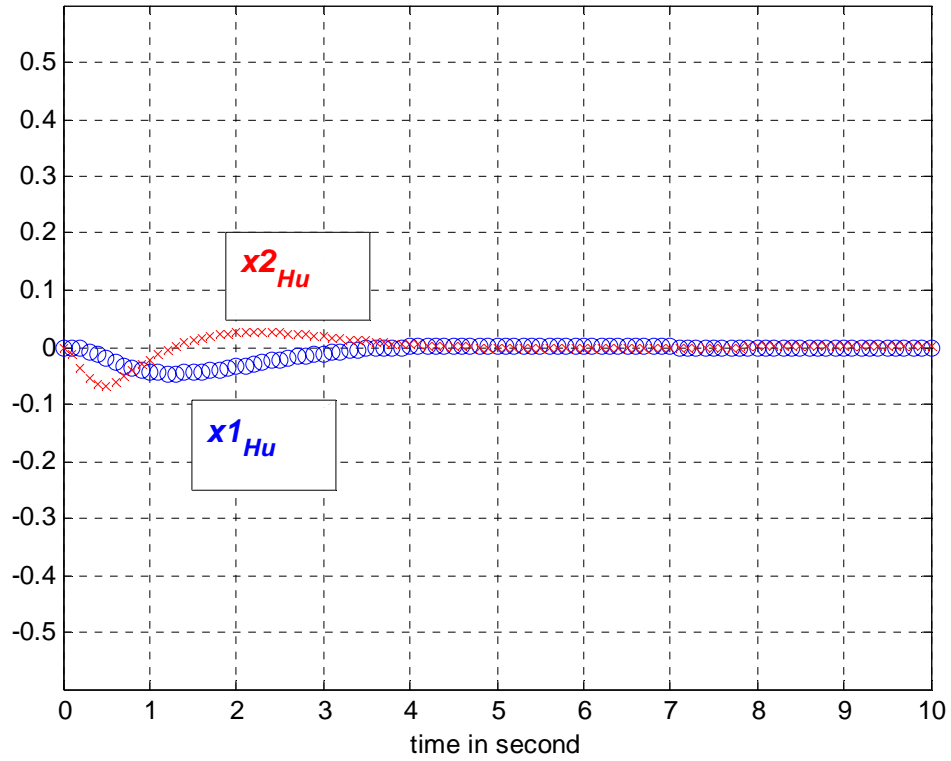
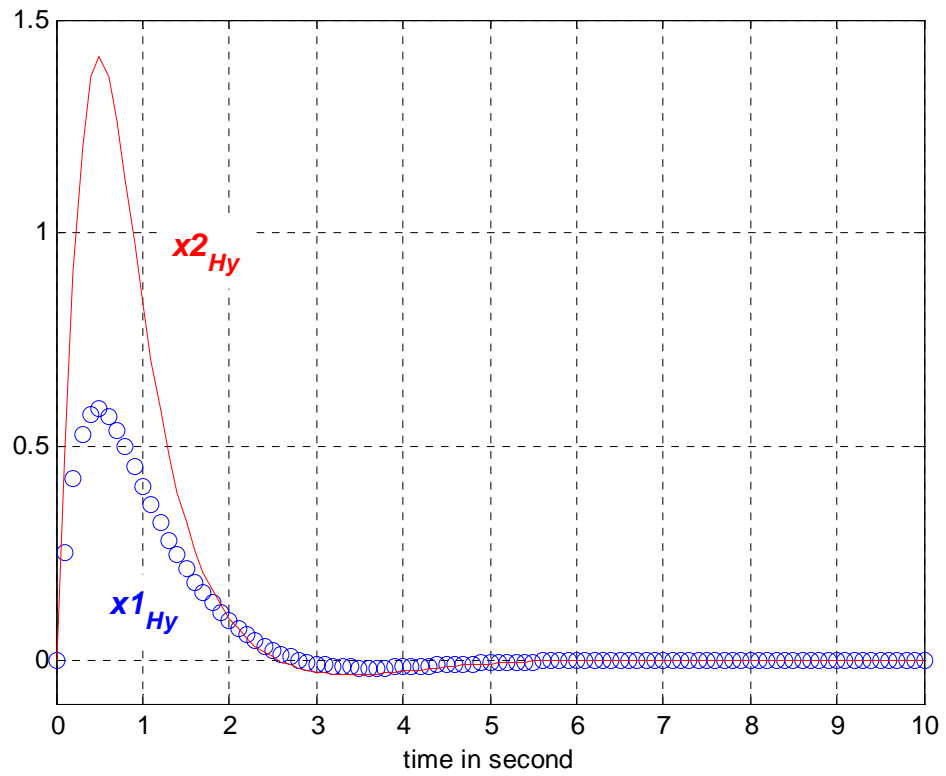Figure 5.3. The zero input responses of the state variables in H_u block in combined observer-controller in observability form
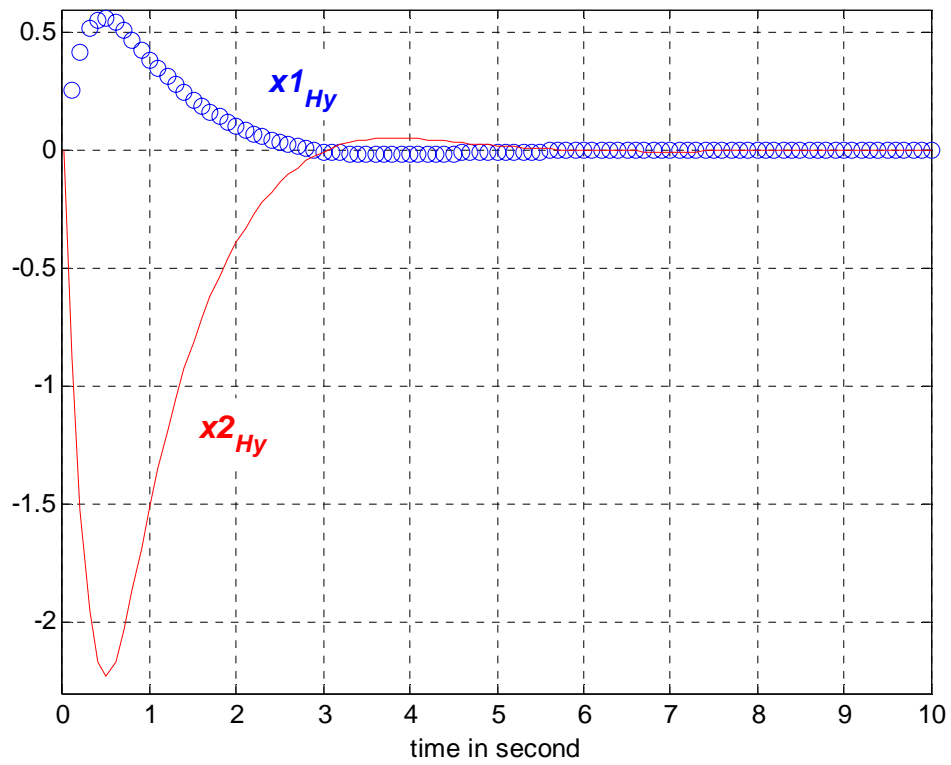
Figure 5.4. The zero input responses of the state variables in H_y block in combined observer-controller in observability form
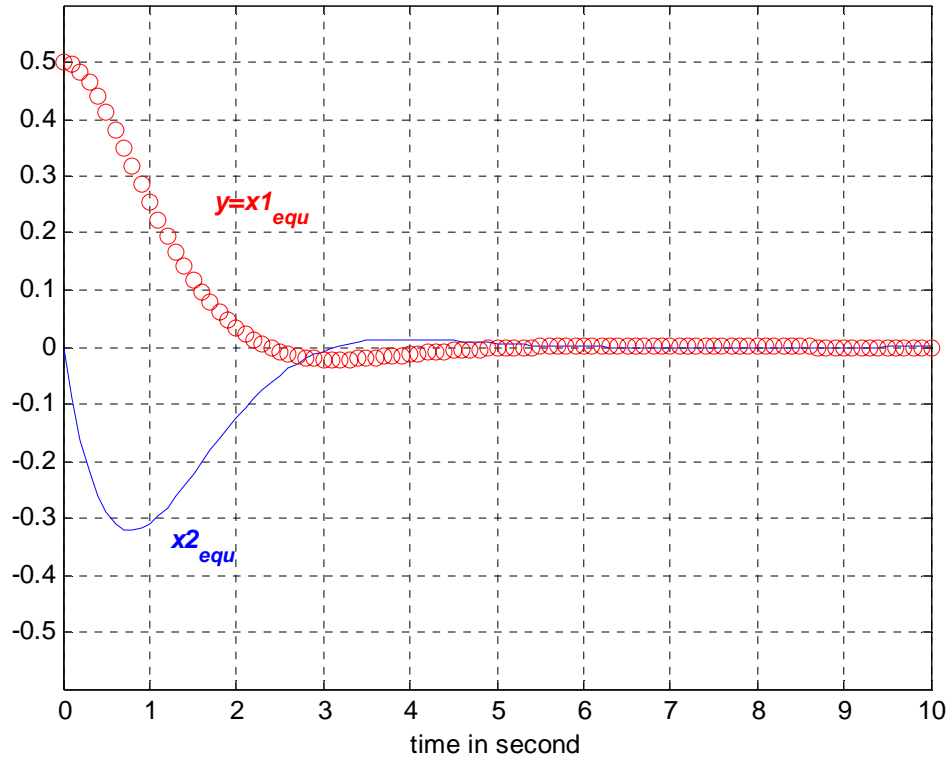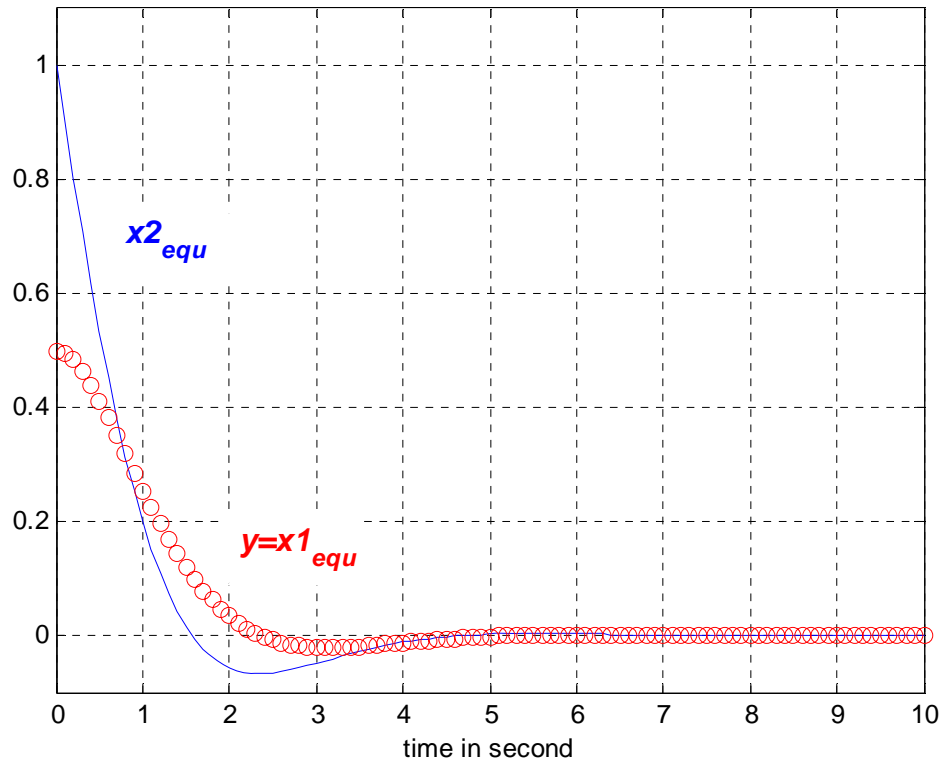
Figure 5.5. The zero input responses of the equivalent overall combined observer-controller in observability form with initial conditions: $y(0) = 0.5$ and $\dot{y}(0) = 0$
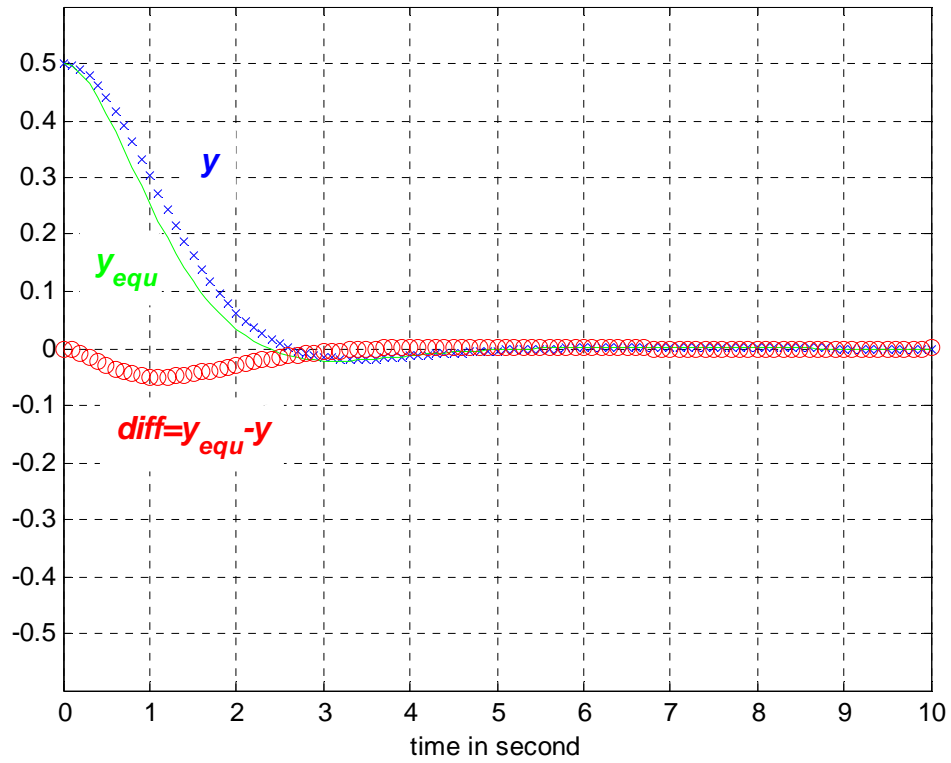
Figure 5.6. The differences of responses between the combined observer-controller in observability form and its equivalent overall system with initial conditions: $y(0) = 0.5$ and $\dot{y}(0) = 0$

Figure 5.7. The zero input responses of the combined observer-controller in observer form with initial conditions: $y = 0.5$ and $\dot{y} = 0$

Figure 5.8. The zero input responses of the plant state variables in combined observer-controller in observer form with initial conditions: $x_{1\_plant}(0) = 0.5$ and $x_{2\_plant}(0) = 1.0$

Figure 5.9. The zero input responses of the state variables in H_u block in combined observer-controller in observer form

Figure 5.10. The zero input responses of the state variables in H_y block in combined observer-controller in observer form

Figure 5.11. The zero input responses of the equivalent overall combined observer-controller in observer form with initial conditions: $y(0) = 0.5$ and $\dot{y}(0) = 0$

Figure 5.12. The differences of responses between the combined observer-controller in observer form and its equivalent overall system with initial conditions: $y(0) = 0.5$ and $\dot{y}(0) = 0$
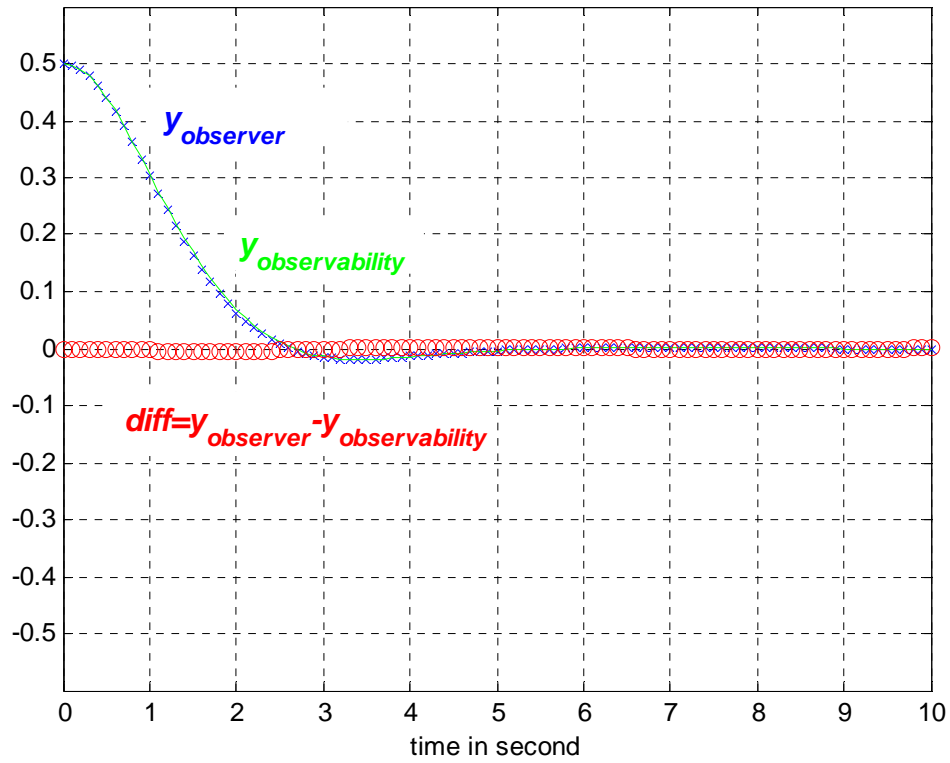
Figure 5.13. The differences of output responses between the observability and observer forms initial conditions: $y(0) = 0.5$ and $\dot{y}(0) = 0$
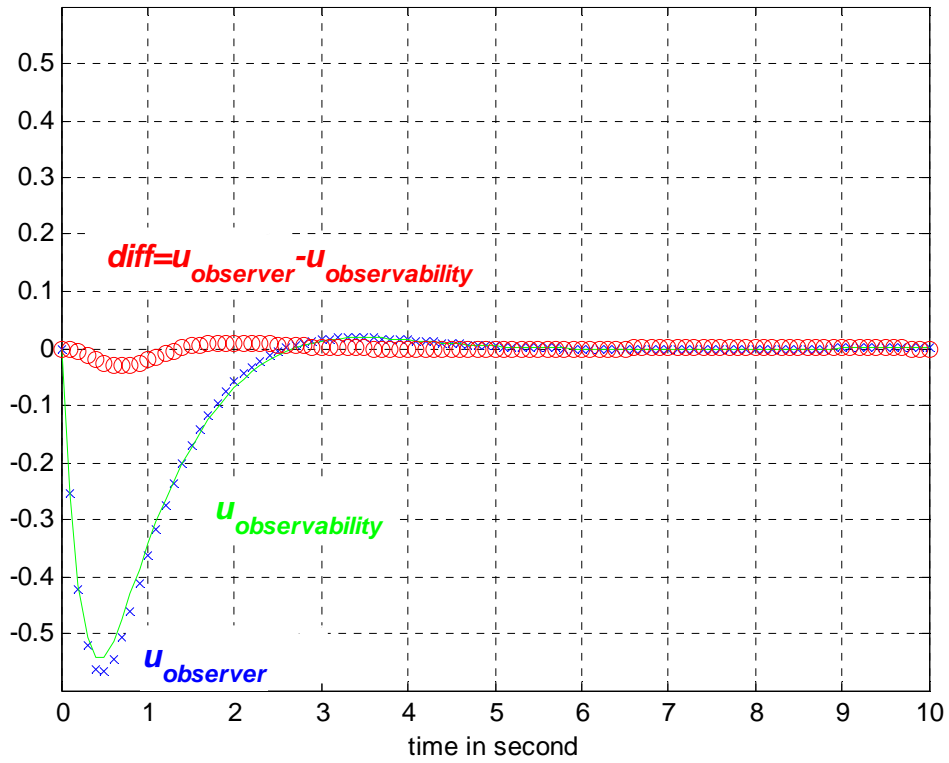
Figure 5.14. The differences of responses of the input between the observability and observer forms initial conditions: $y(0) = 0.5$ and $\dot{y}(0) = 0$
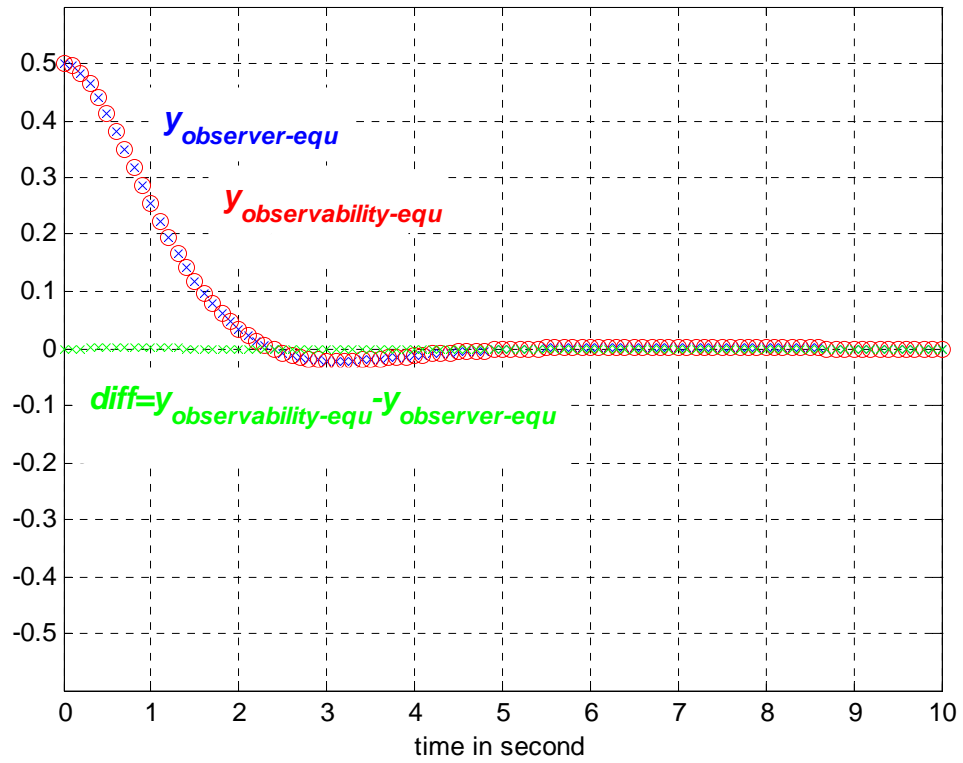
Figure 5.15. The differences of responses between the two equivalent overall combined observer-controller systems with initial conditions: $y(0) = 0.5$ and $\dot{y}(0) = 0$
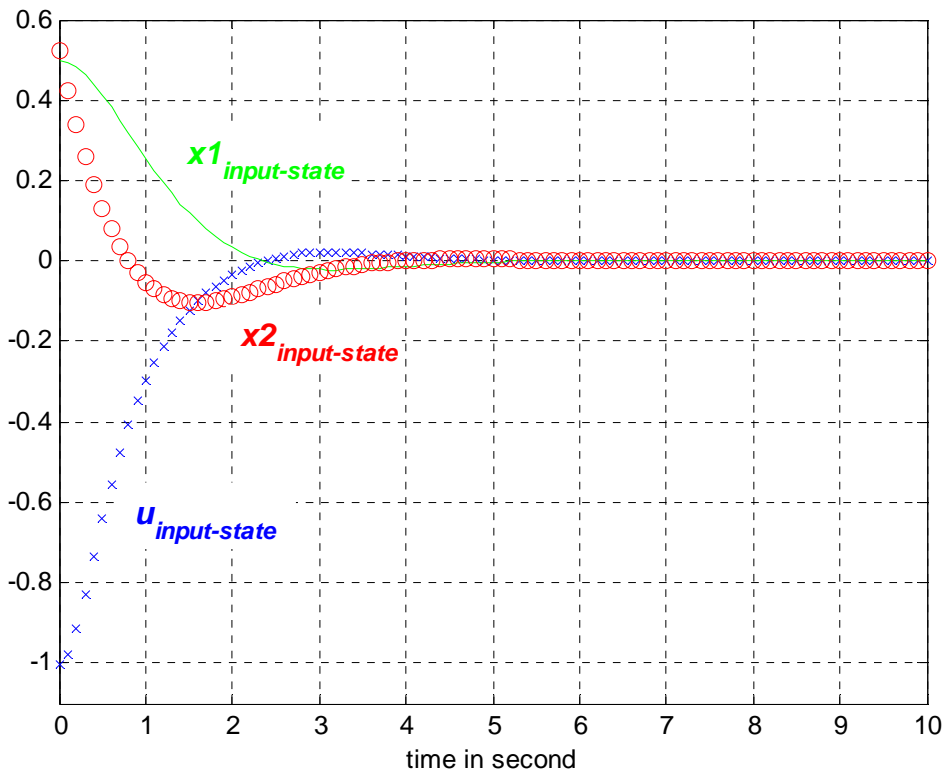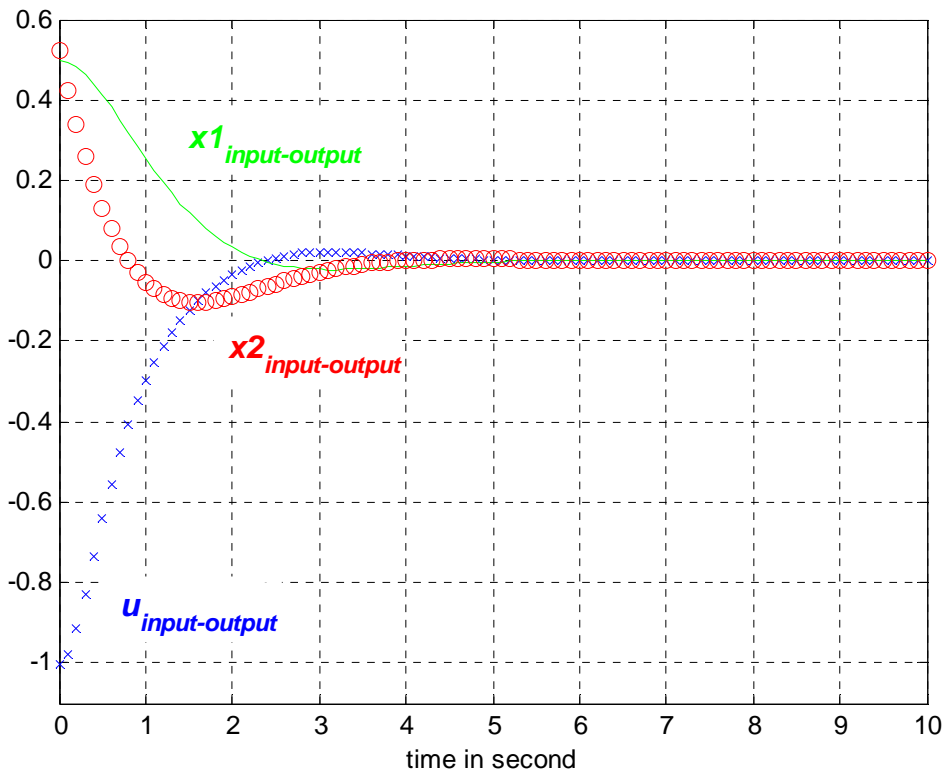
Figure 5.16. The responses of the closed-loop control system using input-state method with initial conditions: $x_{1\_input-state}(0) = 0.5$ and $x_{2\_input-state}(0) = 0.5205744614$

Figure 5.17. The responses of the closed-loop control system using input-output method with initial conditions: $x_{1\_input-output}(0) = 0.5$ and $x_{2\_input-output}(0) = 0.5205744614$
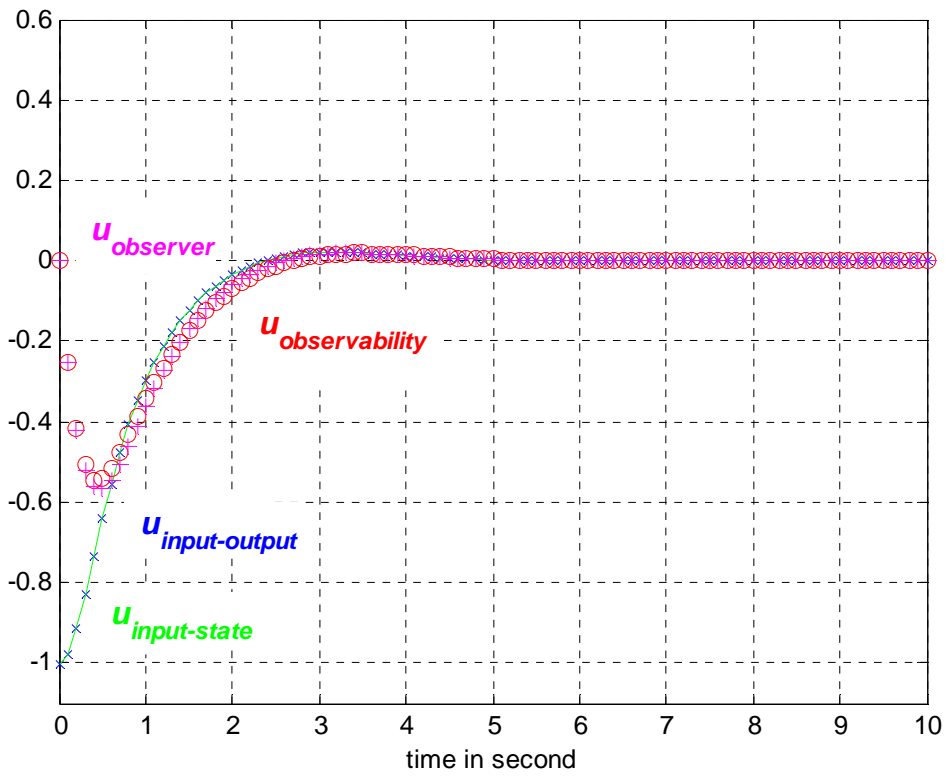
Figure 5.18. The signal responses of the inputs *u* from four methods: combined observer-controller in observability and observer forms, input-state and input-output
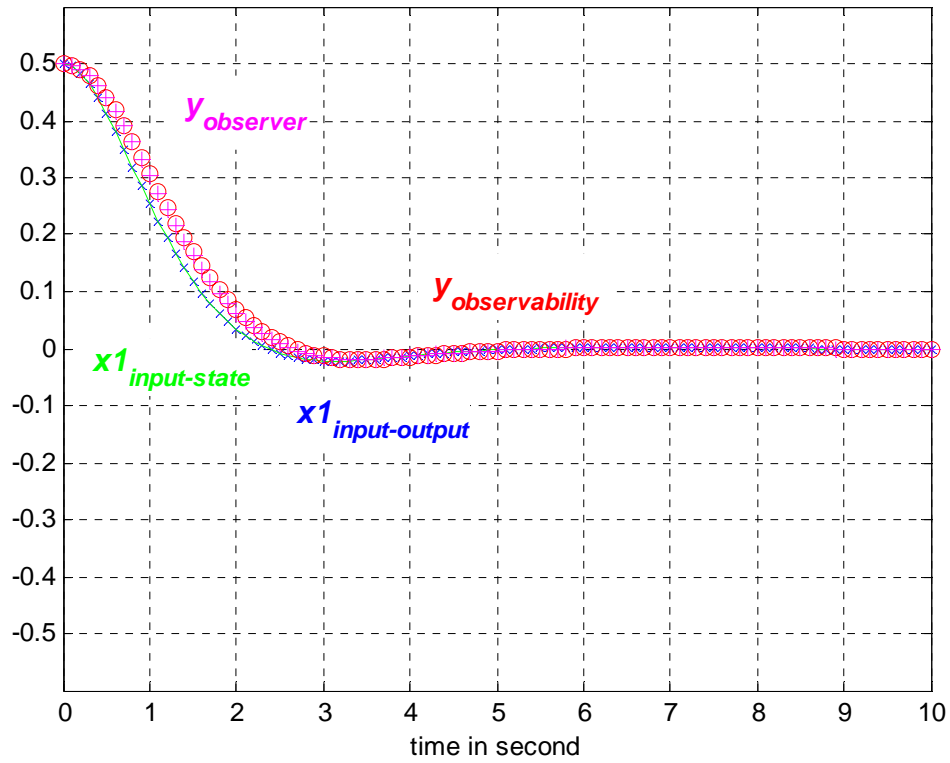
Figure 5.19. The output responses from four methods: combined observer-controller in observability and observer forms, input-state and input-output
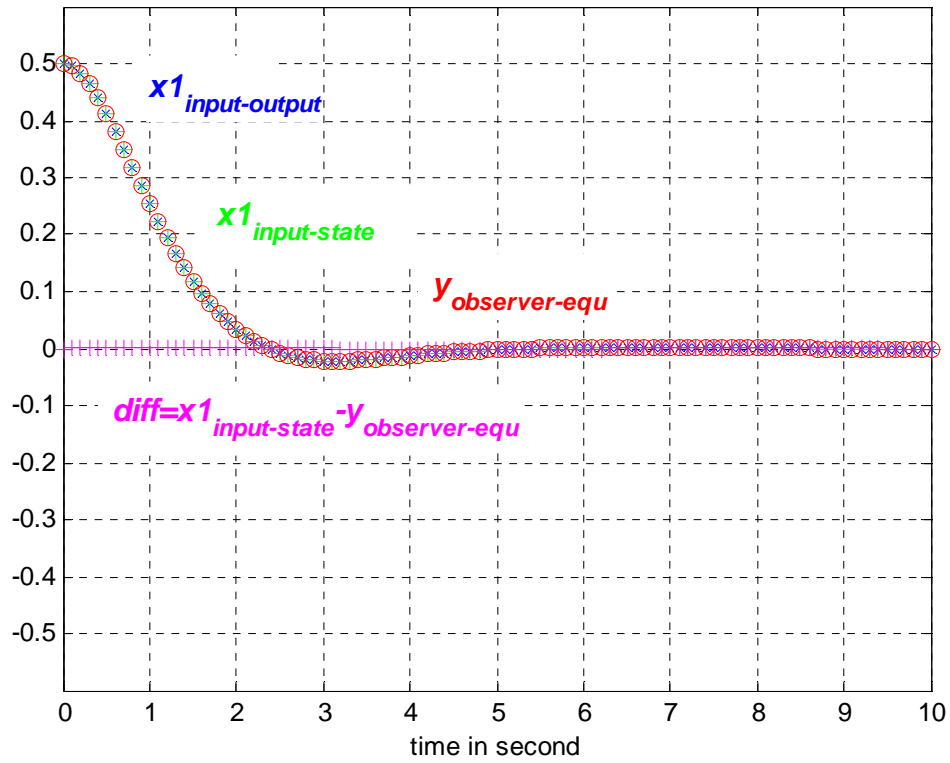
Figure 5.20. The output responses from the equivalent overall combined observer-controller in observer forms, input-state and input-output
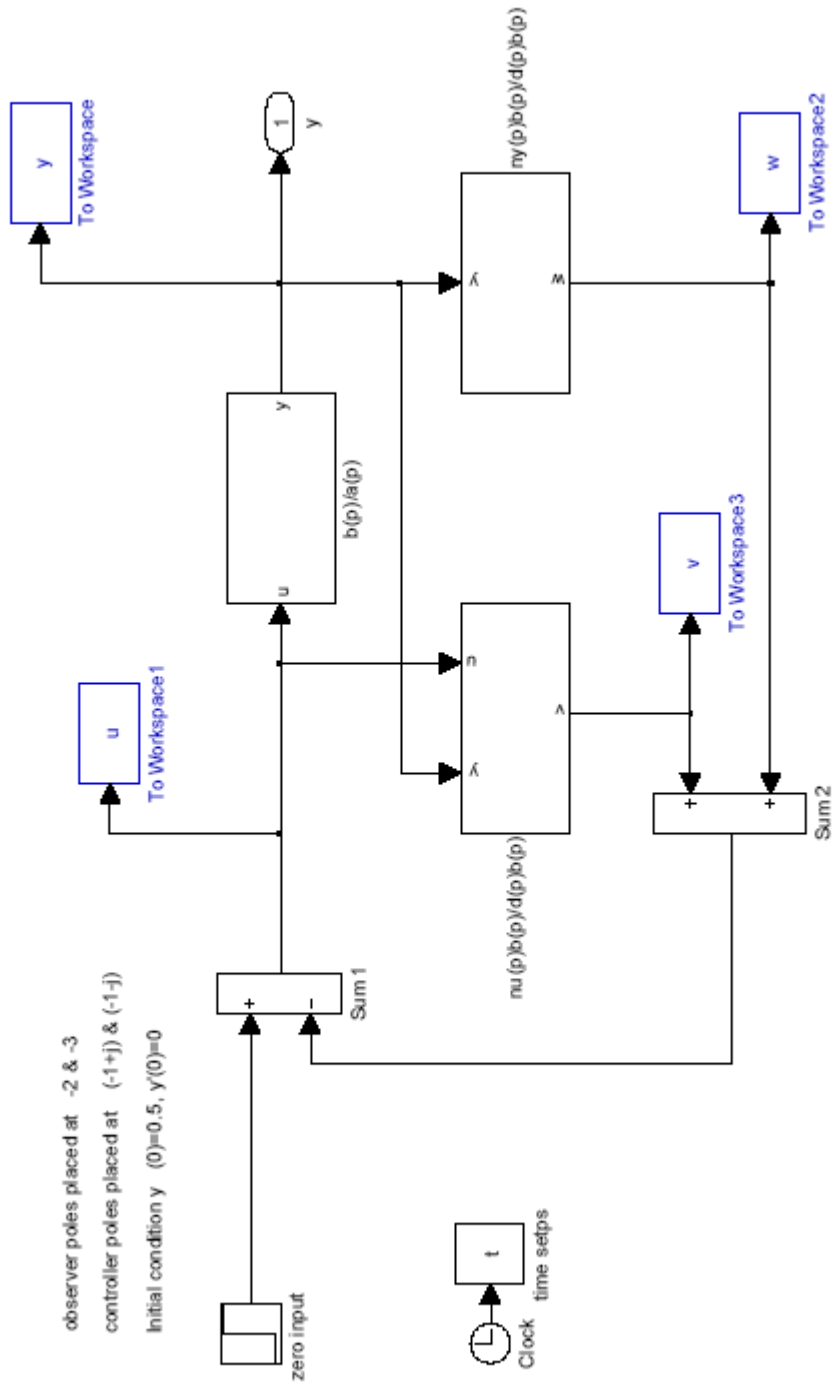
CHAPTER 6

CONCLUSION

A new recursive method to reduce the order of Diophantine equation is presented. In this method, only inversions of triangular matrices are involved. A numerical example is given to illustrate the method. Though this approach does not necessarily offer a computational advantage over the traditional Sylvester matrix approach, it opens up a possibility of solving a *multivariable* Diophantine equation in a systematic manner and involving only the inversion of block triangular matrices.

The setting up of state-variable description for linear time-varying single-input single-output systems from an input-output description is also considered. The derivation of two canonical forms, the observer and the observability canonical forms, are presented. Formulas for the associated initial condition conversion are given. Block diagrams for the realization of the state-variable description are the same as those of the linear time-invariant case, except that the coefficients are time-varying.
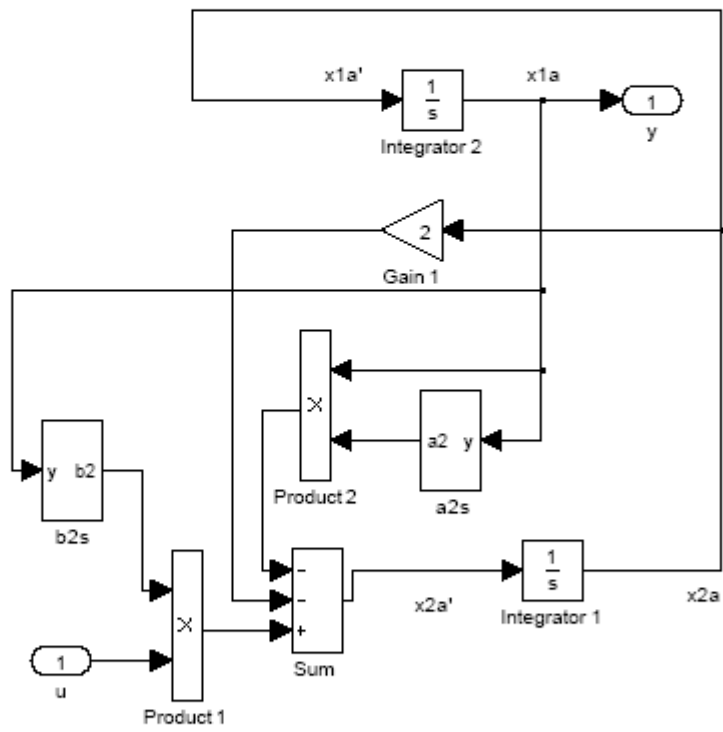
APPENDIX A

SIMULINK DIAGRAM OF COMBINED OBSERVER-CONTROLLER WITH
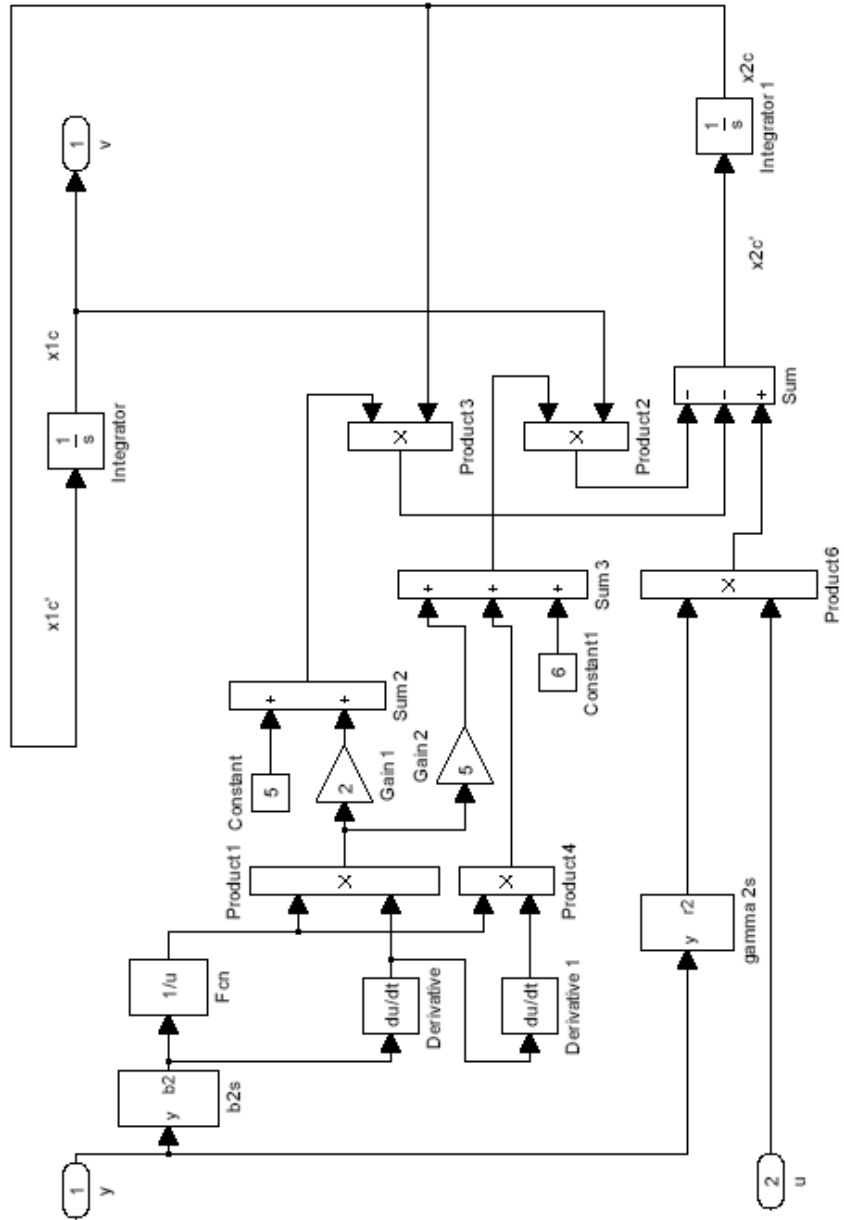OBSERVABILITY CANONICAL FORM IN SECTION 5.1.1

To Workspace

y

1
y

my(p)b(p)/d(p)b(p)

To Workspace2

w

b(p)/a(p)

To Workspace3

v

To Workspace1

u

u

y

v

Sum 2

nu(p)b(p)/d(p)b(p)

Sum 1

observer poles placed at   -2 & -3

controller poles placed at   (-1+j) & (-1-j)

Initial condition y (0)=0.5, y'(0)=0

zero input

Clock

time setps

t

APPENDIX B

SIMULINK DIAGRAM OF SUBSYSTEM PLANT IN COMBINED OBSERVER-
CONTROLLER WITH OBSERVABILITY CANONICAL FORM IN SECTION 5.1.1

x1a'
$\frac{1}{s}$
Integrator 2
x1a

1
y

2
Gain 1

×
Product 2

a2 y
a2s

y b2
b2s

×
Product 1

1
u

−
−
+
Sum

x2a'

$\frac{1}{s}$
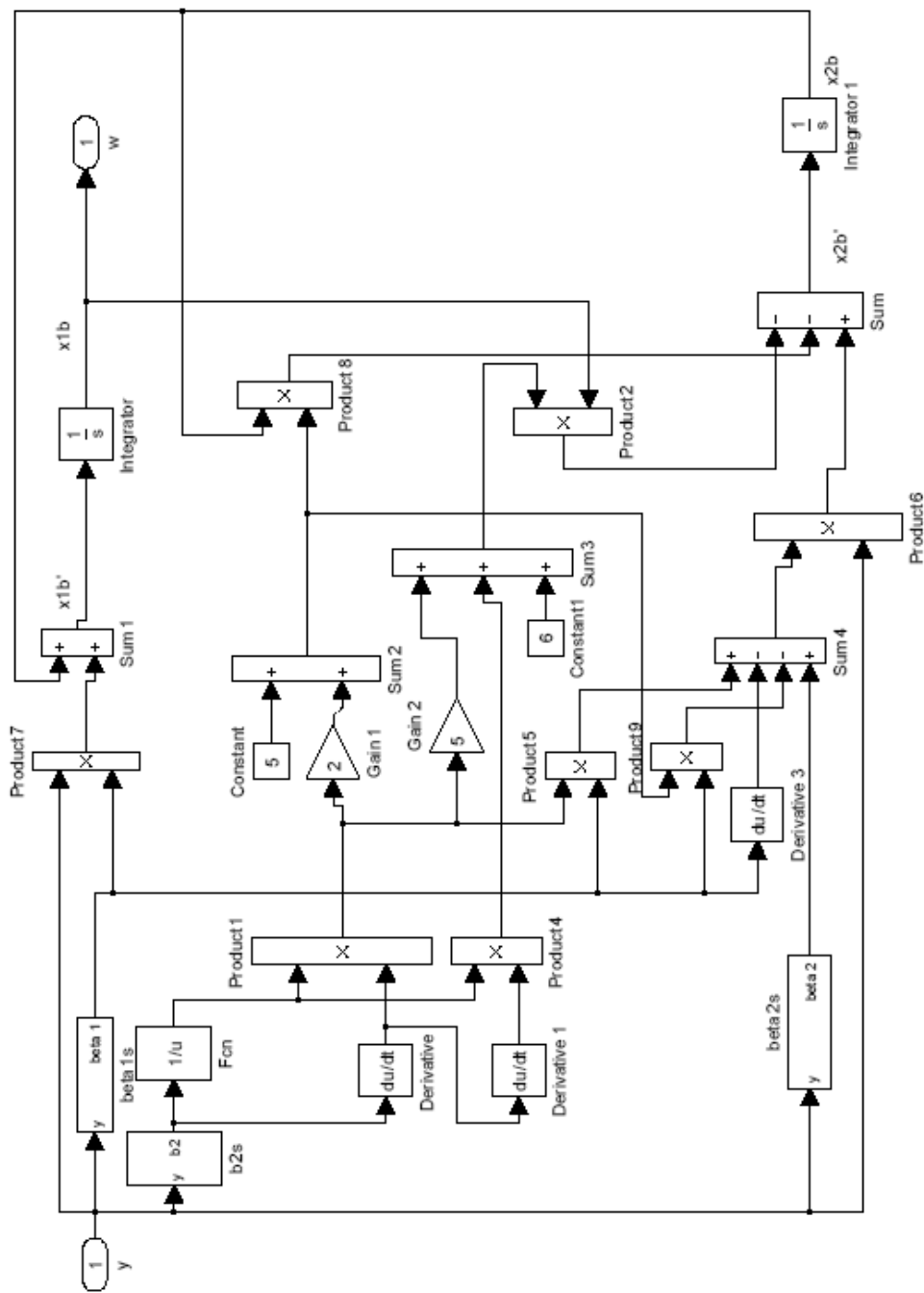Integrator 1

x2a

APPENDIX C

SIMULINK DIAGRAM OF SUBSYSTEM Hu IN COMBINED OBSERVER-
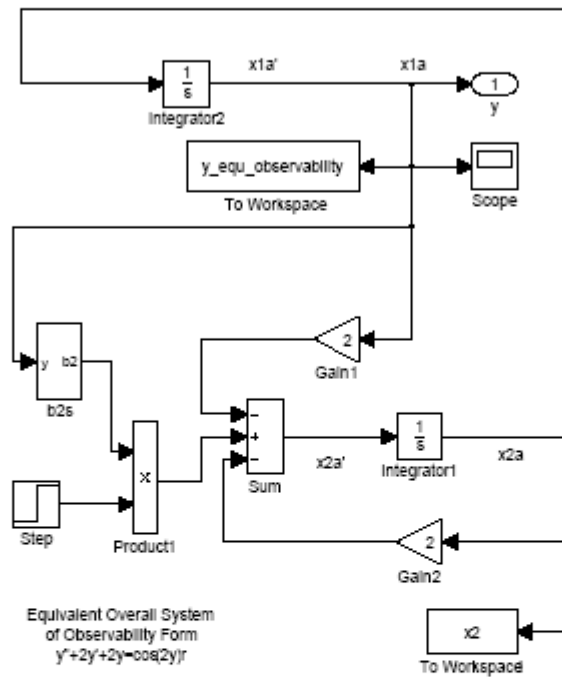CONTROLLER WITH OBSERVABILITY CANONICAL FORM IN SECTION 5.1.1

103

APPENDIX D

SIMULINK DIAGRAM OF SUBSYSTEM Hy IN COMBINED OBSERVER-
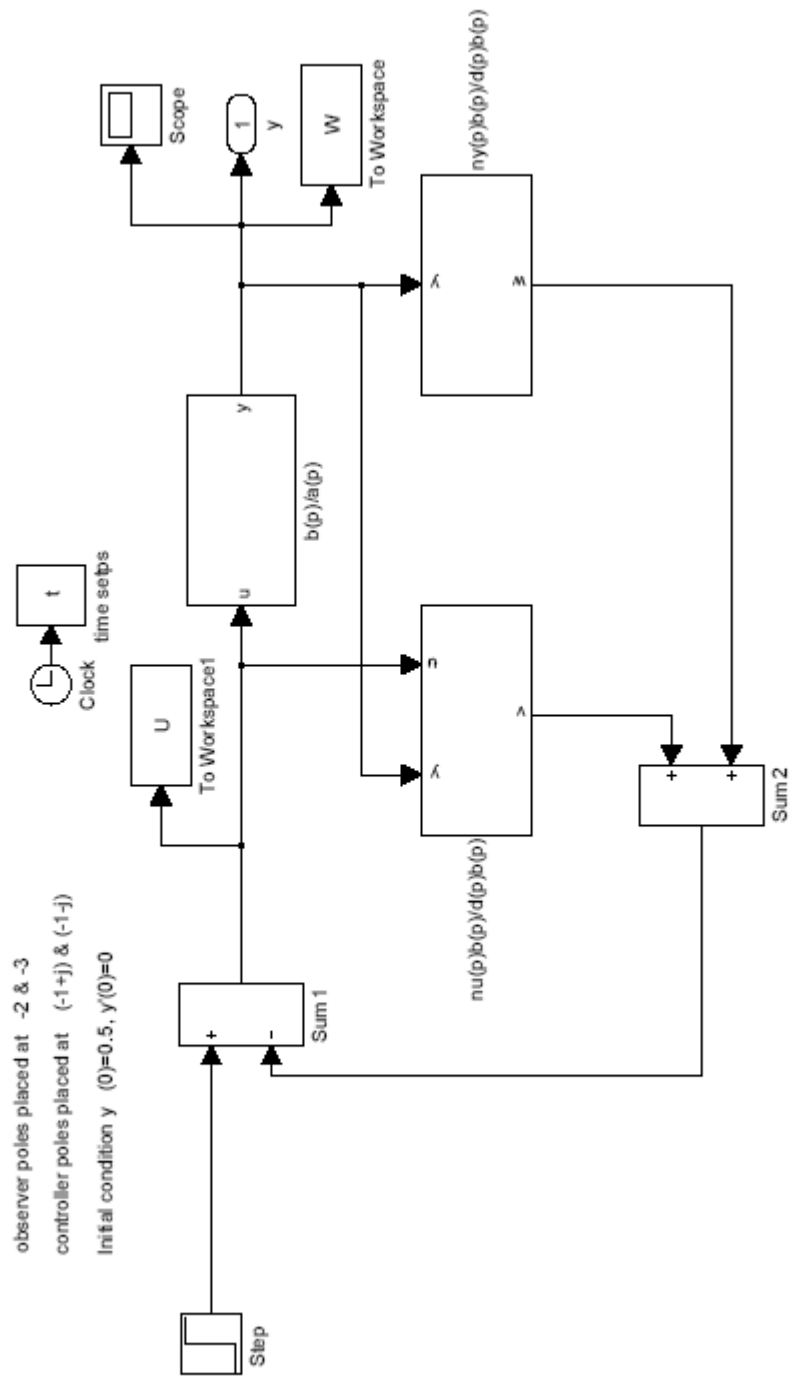CONTROLLER WITH OBSERVABILITY CANONICAL FORM IN SECTION 5.1.1

APPENDIX E

SIMULINK DIAGRAM OF EQUIVALENT OVERALL COMBINED OBSERVER-
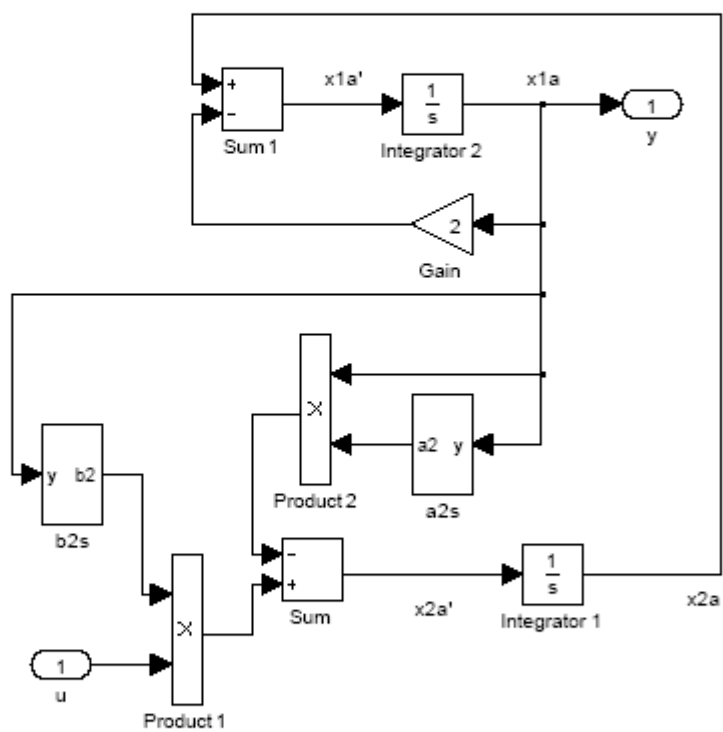CONTROLLER WITH OBSERVABILITY CANONICAL FORM IN SECTION 5.1.1

Equivalent Overall System
of Observability Form
y"+2y'+2y=cos(2y)r

APPENDIX F

SIMULINK DIAGRAM OF COMBINED OBSERVER-CONTROLLER WITH
OBSERVER CANONICAL FORM IN SECTION 5.1.2

observer poles placed at   -2 & -3

controller poles placed at   (-1+j) & (-1-j)

Initial condition y (0)=0.5, y'(0)=0

Scope

y

W

To Workspace

ny(p)b(p)/d(p)b(p)

b(p)/a(p)

t

Clock

time setps

U

To Workspace1

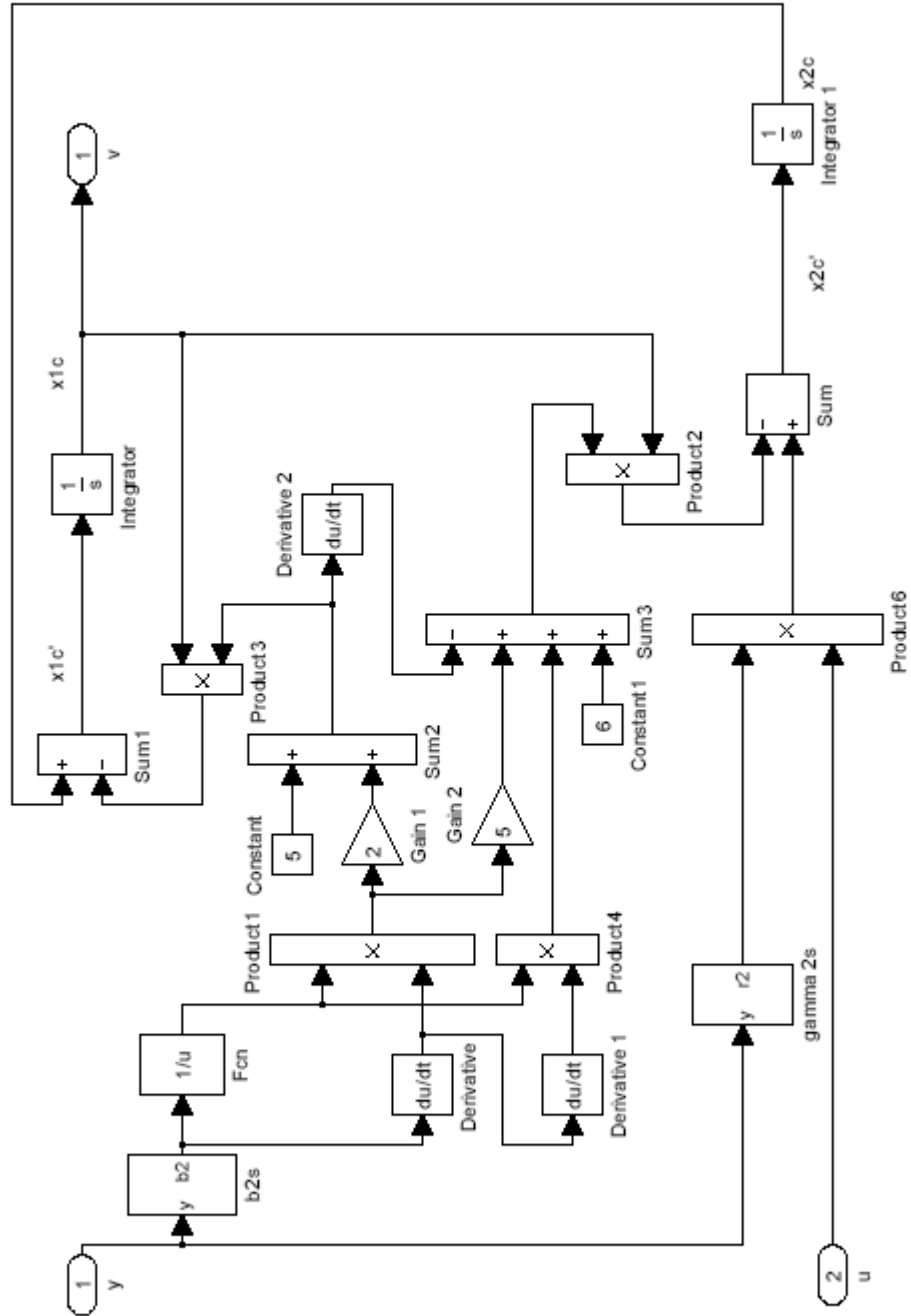nu(p)b(p)/d(p)b(p)

Sum 1

Sum 2

Step

APPENDIX G

SIMULINK DIAGRAM OF SUBSYSTEM PLANT IN COMBINED OBSERVER-
CONTROLLER WITH OBSERVER CANONICAL FORM IN SECTION 5.1.2
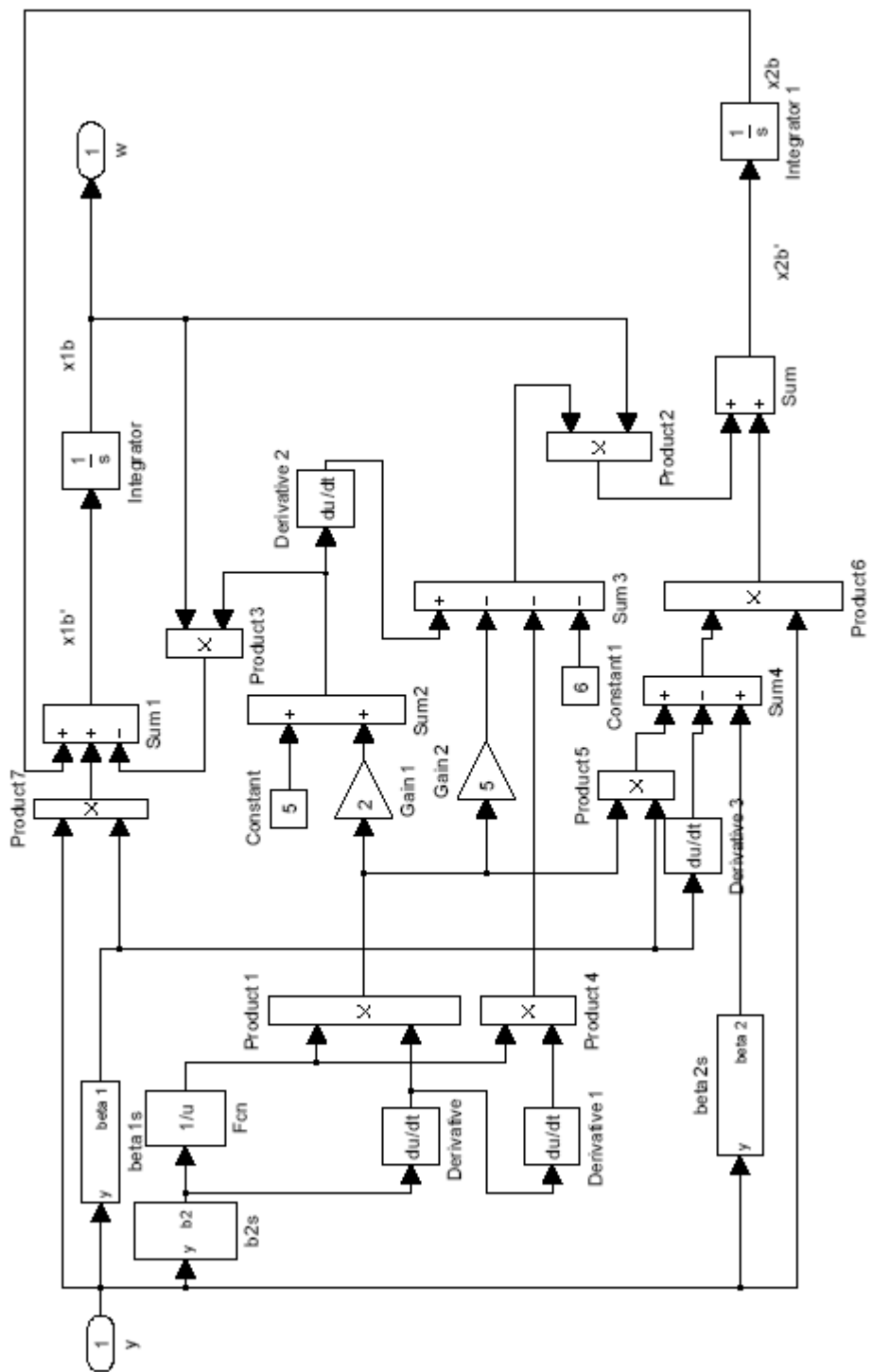
APPENDIX H

SIMULINK DIAGRAM OF SUBSYSTEM Hu IN COMBINED OBSERVER-
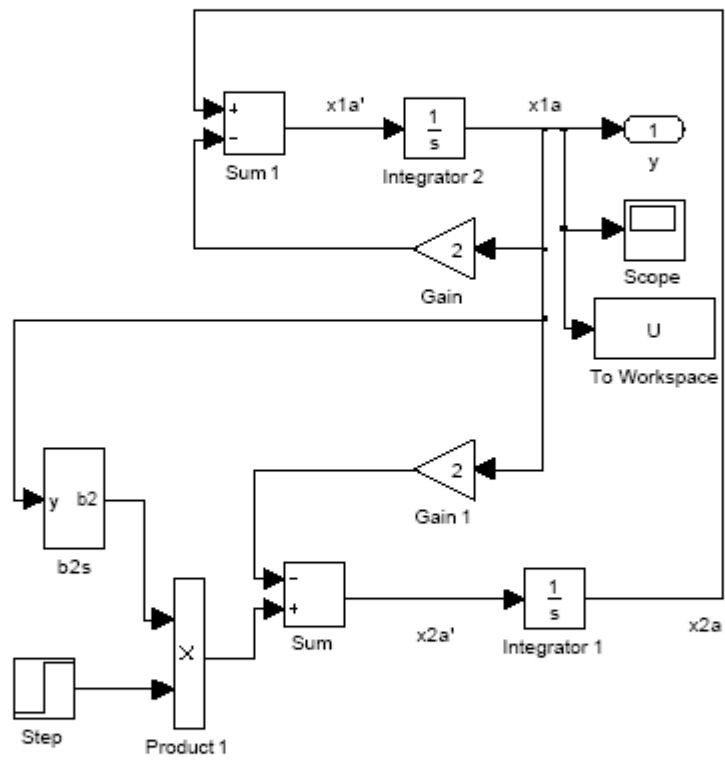CONTROLLER WITH OBSERVER CANONICAL FORM IN SECTION 5.1.2

113

APPENDIX I

SIMULINK DIAGRAM OF SUBSYSTEM Hy IN COMBINED OBSERVER-
CONTROLLER WITH OBSERVER CANONICAL FORM IN SECTION 5.1.2

APPENDIX J

SIMULINK DIAGRAM OF EQUIVALENT OVERALL COMBINED OBSERVER-
CONTROLLER WITH OBSERVER CANONICAL FORM IN SECTION 5.1.2

Sum 1   x1a'   Integrator 2   x1a   y

Gain   Scope

To Workspace

b2s   Gain 1

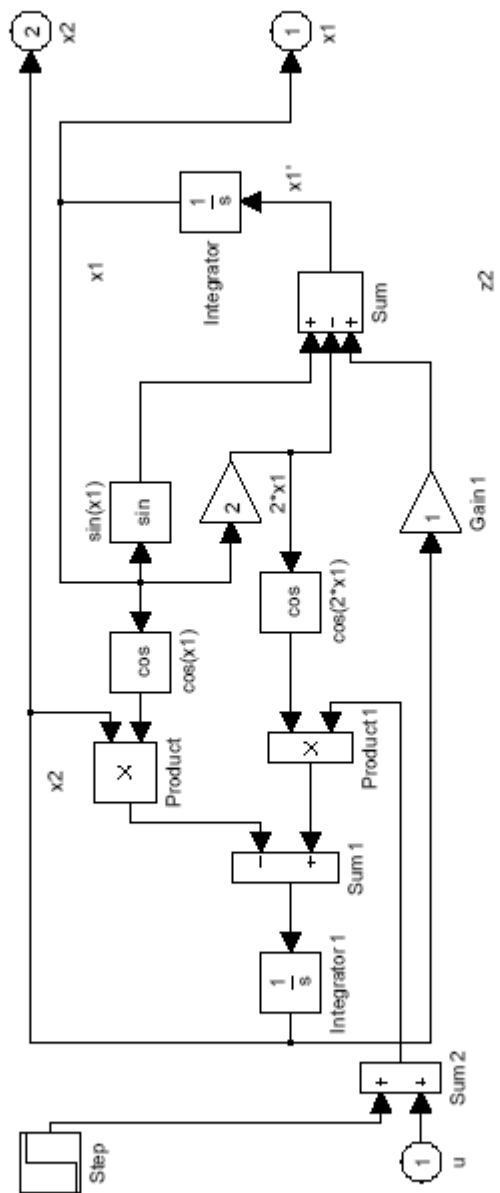Step   Product 1   Sum   x2a'   Integrator 1   x2a

APPENDIX K

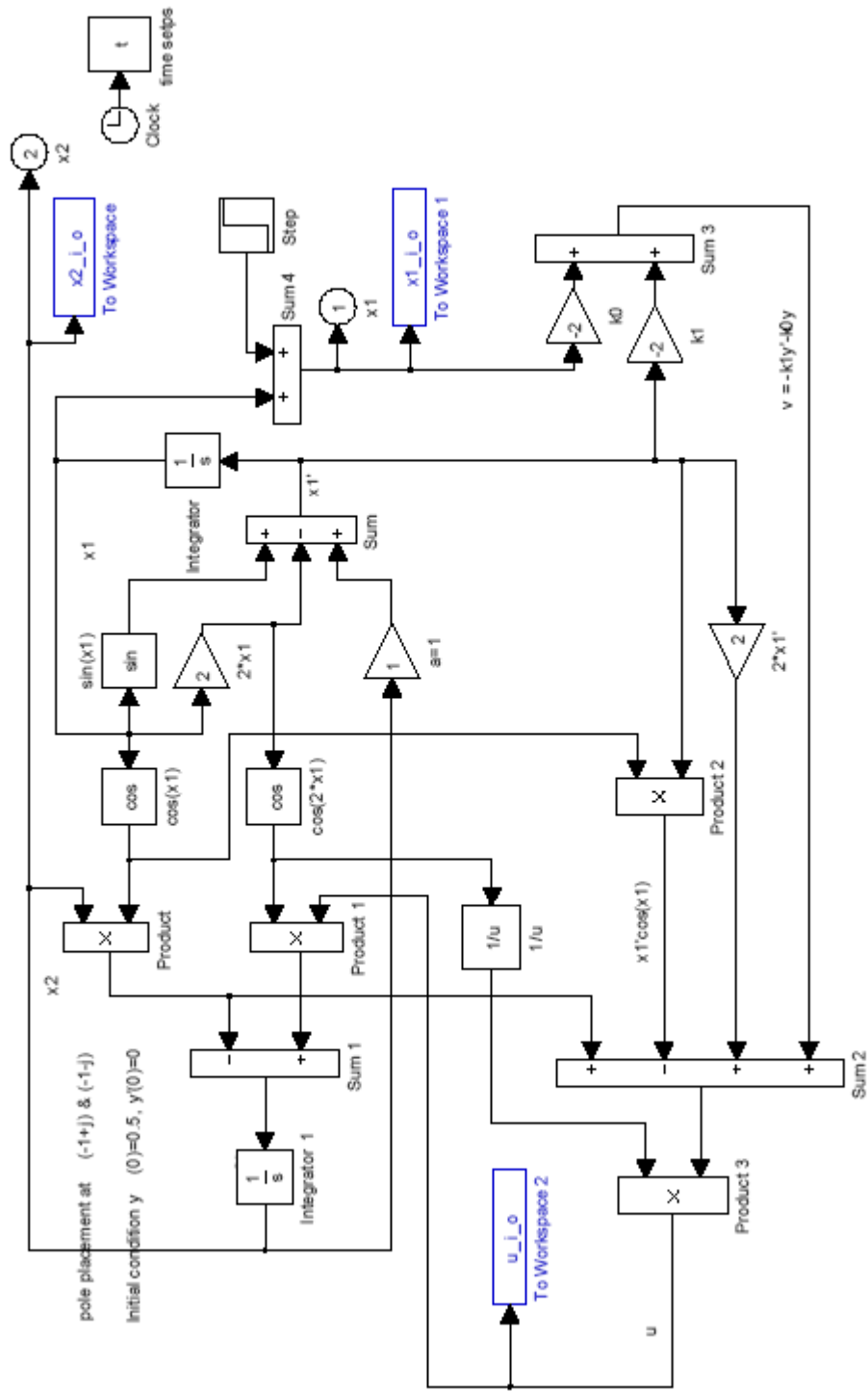SIMULINK DIAGRAM OF INPUT-STATE LINEARIZATION IN SECTION 5.2
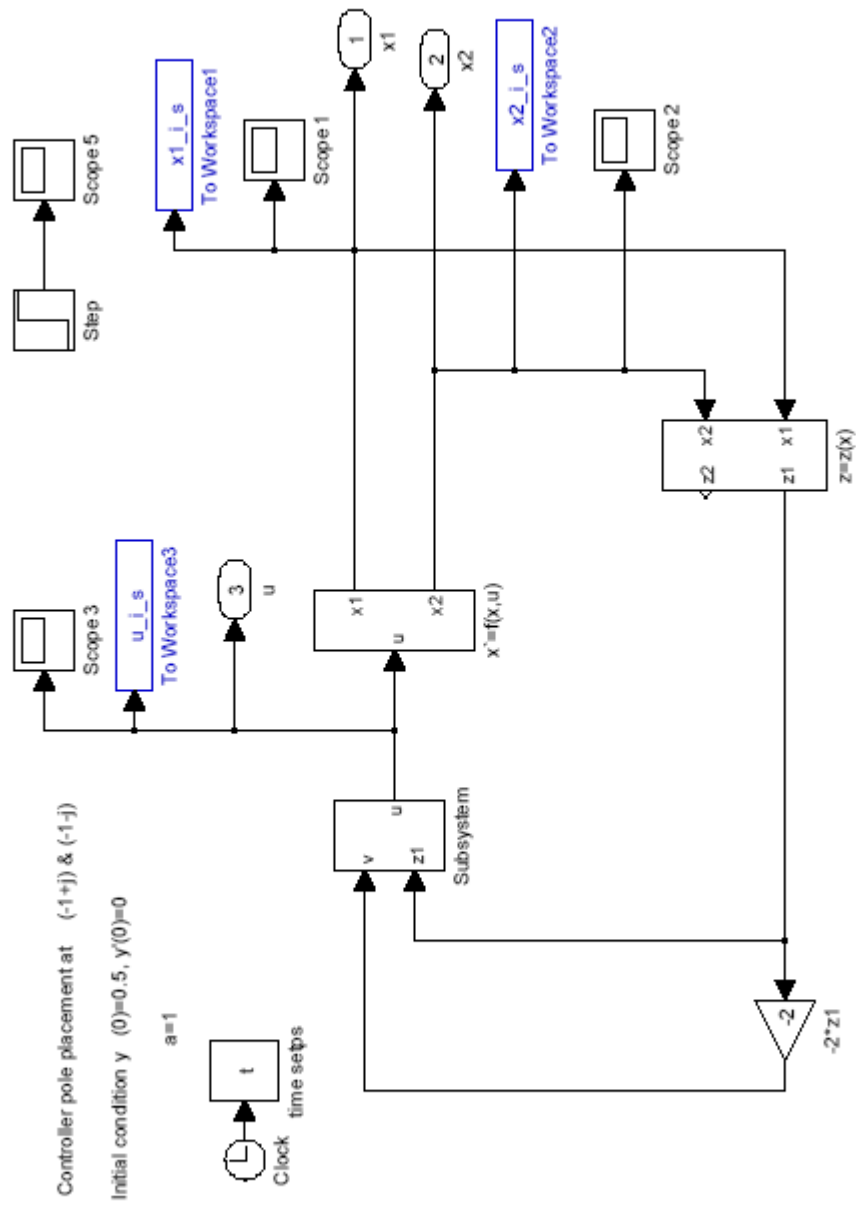
APPENDIX L

SIMULINK DIAGRAM OF SUBSYSTEM OF INPUT-STATE LINEARIZATION IN
SECTION 5.2

APPENDIX M

SIMULINK DIAGRAM OF INPUT-OUTPUT LINEARIZATION IN SECTION 5.3

123

REFERENCES

[1] T. Kailath, Linear systems, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[2] F.M. Callier and C.A. Desoer, Multivariable feedback systems, New York: Springer-Verlag, 1982.

[3] P. J. Antsaklis, *Proper stable transfer matrix factorizations and internal system descriptions*, IEEE Transactions on Automatic Control, vol. 31, no. 7, pp. 634-638, July 1986.

[4] H. Cohen. A Course in Computational Number Theory, Springer, 1993.

[5] M. Newman. Integral Matrices. Academic Press (New York), 1972.

[6] P. Gibbons. Computational methods in design theory. In The CRC Handbook of Combinatorial Designs, edited by C. Colbourn and J. Dinitz, pp. 718-728. CRC Press, 1996.

[7] C.T. Chen, Linear System Theory and Design, Oxford University Press, NY, 1999.

[8] C.H. Fang, *A new method for solving the polynomial generalized Bezout identity*. IEEE Trans. Circuits and Systems, Vol. 39, No. 1, pp.63-65, 1992.

[9] C.H. Fang, *A simple approach to solving Diophantine equation*. IEEE Trans. on Automatic Control, 37, pp. 152-155 , 1992.

[10] K.S. Yeung, T.M. Chen, *Solving matrix Diophantine equation inverting a square nonsingular system of equations*. IEEE Trans. Circuits and Systems, Vol. 51, No. 9, pp. 488-495, 2004.

[11] K.S. Yeung, S. Wu and F. Najjar, *Diophantine equation order reduction using Euclidean Algorithm.* Journal of Computational Mathematics and Optimization, Vol. 3, No. 2, pp. 115-128, May 2007.

[12] Michael Alekhnovich, *Linear Diophantine equations over polynomials and soft decoding of Reed-Solomon codes*, Proceeding of 43rd annual IEEE Symposium on Foundations of Computer Science(FOCS'02).

[13] K. H. Plarre, A M Rojas, and R A Rojas, *Analytical solution to the Polynomial Diophantine Equation: Development and Application to Generalized Predictive Control Analysis,* UKACC International Conference on Control '98, pp. 804-809, 1-4 Sep. 1998.

[14] Feng Kuo, Chi-Tsong Chen, *A Recursive Algorithm for Coprime Fractions and Diophantine Equation*, IEEE Transactions on Automatic Control, vol. 34, no. 12, pp. 1276-1279, Dec. 1989.

[15] E. W. Kamen, P. P. Khargonekar and K. R. Poola, *A transfer-function approach to linear time-varying discrete-time systems*, SIAM J. Control and Optimization, 23, pp. 550-565, 1985.

[16] J. Jezek and I. Nagy, *An algebraic approach to the synthesis of control for linear time-varying systems on a finite time horizon*, Prep. IFAC World, hop on System Structure and Control, Prague, Czechoslovakia, pp. 63-68, 1989.

[17] V. L. Syrmos and F. L. Lewis, *Output feedback eigenstructure assignment using two Sylvester equations*, IEEE Transactions on Automatic Control, vol. 38, no. 3, pp. 495-499, March 1993.

[18] S. P. Bhattacharrya and E. de Souza, *Pole assignment via Sylvester equation,* System Control Letter, vol. 1, pp. 261-263, 1982.

[19] R. K. Cavin, III and S. P. Bhattacharrya, *Robust and well-conditioned eigenstructure assignment via Sylvester equation*, in Proc. Amer. Contr. Conf., Arlington, VA, pp. 1053-1057, 1982.

[20] P. Van Dooren, *A reduced-order observer: A new algorithm and proof*, System Control Letter vol. 4, pp. 243-250, 1984.

[21] C. C. Tsui, *An algorithm for computing state feedback in multi-input linear systems*, IEEE Transactions on Automatic Control, vol. AC-31, no. 3, pp. 243-246, March 1986.

[22] K. S. Tsakalis and P. A. Ioannou, Linear Time-Varying Systems, Prentice-Hall, N.J., pp.19-25, 1993.

[23] W. A. Wolovich, *A Division Algorithm for Polynomial Matrices, IEEE Transactions on Automatic Control*, vol. AC-29. no. 7, pp.656-658, July 1984.

[24] J. S. H. Tsai and S. S. Chen, *Generalized Matrix Euclidean Algorithms for Solving Diophantine Equations and Associated Problems*, Computers & Mathematics Applications, vol. 25, no. 12, pp. 755-757, June 1993.

[25] H. H. Rosenbrock, State-Space and Muhivariable Theory, Nelson, London, 1970.

[26] W. A. Wolovich, Linear Multivariable System. New York: Spnger-Verlag, 1974.

[27] R. R. Bitmead, *S.-Y.* Kung, B. D. O. Anderson and T. Kailath, *Greatest common divisors via generalized Sylvester and Bezout matrices,* IEEE Transactions on Automatic Control, vol. AC-23, no. 6, pp. 1043-1047, Dec. 1978.

[28] W. Thomson, *Mechanical Integration of the General Linear Differential Equation of Any Order with Variable Coefficients, Proc. Roy. Soc.*, vol. 24, pp.271-275, 1876.

[29] K. Ogata, State space analysis of control systems, Prentice-Hall, N.J., 1987

[30] M. Zeitz, *Observability canonical (phase-variable) form for non-linear time variable systems*, International Journal of Systems Science, vol.15, no. 9, Sep. 1984, pp. 949-958.

[31] A. J. Krener, A. Isidori, *Linearization by output injection and nonlinear observers Systems & control letters,* vol.3, no. 1, pp.47-52, June 1983.

[32] A. J. Schaft, *Representing a nonlinear state space system as a set of higher-order differential equations in the inputs and outputs,* Systems & Control letters, vol.12, pp. 151-160, 1989.

[33] J. A. Heinen, *The relationship of initial conditions of a scalar differential equation to those of its state variable representation,* International Journal of Control, vol. 31, no. 4, pp. 705-707, 1980.

[34] D. Knuth, The Art of Computer Programming, Vol. II: Seminumerical Algorithms, Addison-Wesley, Reading, Mass., P.294, 1969.

[35] J. E. Slotine, W. Li, Applied Nonlinear Control, Prentice-Hall, pp. 213, N.J., 1991.

BIOGRAPHICAL INFORMATION

Shr-Hua Wu, also known as Joshua Wu, was born in Taoyuan, Taiwan in 1965. He received the B.S. degree in Electrical Engineering from the Feng-Chia University, Taiwan in 1989, and the M.S. degree in Institute of Optical Sciences from National Central University, Taiwan in 1991. In 1994, he started a doctoral program in Electrical Engineering at The University of Texas at Arlington. He was awarded a Doctor of Philosophy degree in Electrical Engineering in 2007.

From 1992 to 1994, he served as a Lecturer in Electrical Engineering at Fushin Institute of Technology (now named Nangyang Institute of Technology) of Taiwan. From 1998 to 2000, he was a DSP software engineer at Garrett Electronics and built the security Walk-through with digital signal processing technology deployed in Sidney Olympic. From 2000 to 2007, He was a DSP specialist at Nokia. He served as DSP and System Integration chiefs for several projects including Nokia 6560 and E62 smart phones. In April 2007, He joined Research in Motion for developing BlackBerry devices.