Full- and Reduced-Order Linear Observer Implementations in Matlab/Simulink

VERICA RADISAVLJEVIC-GAJIC

The design of observers is usually considered a graduate-level topic and therefore tends to be taught in a graduate-level control engineering course. However, several recent editions of standard undergraduate control-system textbooks cover full-order, and even reduced-order, observers [1]–[9]. Observers are also used in their own right to strictly observe the state variables of a dynamic system rather than to be used for feedback control (for example, in an experiment whose state variables have to be monitored, observed, or estimated at all times).

This tutorial is primarily for undergraduate students and their instructors as a supplement to textbooks, laboratory manuals, and project design assignments. This tutorial is intended to be a self-contained and complete presentation of full- and reduced-order observer designs drawn from several undergraduate and graduate textbooks. Elementary knowledge of state-space and linear algebra is assumed. In addition, this article attempts to resolve challenges that undergraduate students are faced with while implementing full- and reduced-order observers in Matlab/Simulink. The implementation part of the tutorial demonstrates

» how to input full- and reduced-order observer system, input, and output matrices using Simulink state-space blocks and how to determine dimensions of the observer output matrices (they are either identity or zero matrices)

» how to find proper full- and reduced-order observer feedback gains

» how to set observer initial conditions.

To that end, two novel theoretical results have also been developed: how to set up the reduced-order initial conditions using the least-squares method, derived in (38)–(40), and an observation that the reduced-order observer output is identical to the original system's actual output, the result established in (36).

This tutorial is also useful for practicing engineers and scientists interested in controlling or observing linear dynamic systems. This tutorial has been used, in various forms, by the author and her colleagues over several years at several academic institutions: Rutgers University; Villanova University; California State University, Los Angeles; American University of Sharjah; University of Belgrade; and Lafayette College.

Digital Object Identifier 10.1109/MCS.2015.2449691
Date of publication: 16 September 2015

Hopefully, by understanding full- and reduced-order observer design and Matlab/Simulink implementation, students, instructors, engineers, and scientists will appreciate the importance of observers and feel confident using observers and observer-based controllers in numerous engineering and scientific applications.

In undergraduate classes on control systems, time invariant linear systems are represented, in state-space form, by

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t), \quad x(0) = x_0, \tag{1}$$

where $x(t)$ is the state-space vector of dimension n, $u(t)$ is the system input vector (which may be used as a system control input) of dimension m, and matrices A and B are constant and of appropriate dimensions. In practice, the initial condition is often unknown, in which case an observer is designed, [1]–[9], to estimate or observe system state-space variables at all times.

To take the advantage of the useful features of feedback (see, for example, [10, Chap. 12]), it is often assumed that all state variables are available for feedback (full-state feedback), allowing that a feedback control input can be applied as

$$u(x(t)) = -Fx(t), \tag{2}$$

where F is a constant feedback matrix of dimension $m \times n$. The fact that *all state-space variables must be available for feedback* is a prevalent implementational difficulty of full-state feedback controllers. Moreover, large-scale systems with full-state feedback have many feedback loops, which might become very costly and/or impractical. Moreover, often not all state variables are available for feedback. Instead, an output signal that represents a linear combination of the state-space variables is available

$$y(t) = Cx(t), \tag{3}$$

where $\dim\{y(t)\} = l < n = \dim\{x(t)\}$. It is assumed that $l = c = \text{rank}\{C\}$, so there are no redundant measurements. In such a case, under certain conditions, an observer can be designed that is a dynamic system driven by the system input and output signals with the goal of reconstructing (observing, estimating) all system state-space variables at all times.

This article shows how to implement full- and reduced-order observers using the Matlab and Simulink software packages for computer-aided control system design. In fact, how to implement a linear system and its observer, represented by their state-space forms, using the Simulink state-space blocks is shown. How to choose the observers' initial conditions and how to set the observers' gains is also discussed. This presentation is at the level of undergraduate junior or senior students taking their first undergraduate control systems class and assumes knowledge of the linear state-space system representation and linear algebra.

FULL-ORDER OBSERVER DESIGN

The theory of observers originated in the mid 1960s [11]–[13]. According to [11], any system driven by the output of the given system can serve as an observer for that system. Consider a linear dynamic system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t),\ x(t_0) = x_0 = \text{unknown}, \\ y(t) &= Cx(t). \end{aligned} \tag{4}$$

The system output variables $y(t)$ are available at all times, and that information can be used to construct an artificial dynamic system of the same order (n) as the system under consideration that can estimate the system state-space variables at all times. Since the matrices A, B, C are known, it is rational to postulate an observer for (4) as

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t), \\ \hat{x}(t_0) &= \hat{x}_0, \\ \hat{y}(t) &= C\hat{x}(t). \end{aligned} \tag{5}$$

If the outputs $y(t)$ and $\hat{y}(t)$ are compared, they will, in general, be different since, in the first case, the initial condition of (4) is unknown, and, in the second case, the initial condition of the proposed observer (5) is chosen arbitrarily by a control engineer (designed). The difference between these two outputs generates an error signal

$$y(t) - \hat{y}(t) = Cx(t) - C\hat{x}(t) = Ce(t), \tag{6}$$

which can be used as the feedback signal to the observer such that the estimation (observation) error $e(t) = x(t) - \hat{x}(t)$ is reduced. An observer that takes into account feedback information about the observation error is [11]–[13]

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + K(y(t) - \hat{y}(t)) \\ &= A\hat{x}(t) + Bu(t) + KCe(t), \end{aligned} \tag{7}$$

where the matrix K represents the observer gain, which must be selected such that the observation error tends to zero as time increases. From (4) and (7), comes an expression for dynamics of the observation error

$$\begin{aligned} \dot{e}(t) &= (A - KC)\, e(t), \\ e(t_0) &= \text{unknown}. \end{aligned} \tag{8}$$

If the observer gain K is chosen such that the feedback matrix $A - KC$ is *asymptotically stable* (has all eigenvalues with negative real parts), then the estimation error $e(t)$ will decay to zero for any initial condition $e(t_0)$. *This stabilization requirement can be achieved if the pair* (A, C) *is observable* [14]. It can be also noted that the error (and observer) stabilization can be achieved under a weaker condition that the pair (A, C) is detectable (not all, but at least the unstable modes of matrix A are observable) [14].

A standard rule of thumb is that an observer should be designed such that its response is much faster than the system. This is especially important when the observed state variables are used for the purpose of feedback control. Namely, since the system changes in time, its estimated state variables must be as current as possible, otherwise the feedback signals represent considerably delayed estimates of the actual state variables, which can make the controller inaccurate and inefficient. This can be achieved theoretically by choosing the observer eigenvalues to be about ten times faster than the system eigenvalues, that is by setting the smallest real part of the observer eigenvalues to be ten times larger than the largest real part of the closed-loop system eigenvalues

$$|\operatorname{Re}\{\lambda_{\min}(A - KC)\}|_{\text{observer}} > 10|\operatorname{Re}\{\lambda_{\max}(A - BF)\}|_{\text{system}}. \tag{9}$$

Theoretically, an observer can be made arbitrarily fast by locating its closed-loop eigenvalues very far to the left in the complex plane, but very fast observers can generate noise, which is not desirable. In control system practice and applications, instead of ten times larger, it is sufficient that the closed-loop observer eigenvalues are faster than the closed-loop system eigenvalues by a factor of five to six times.

After the system eigenvalues $\lambda_i(A - BF)$ are determined, the observer eigenvalues $\lambda_i(A - KC)$ are placed in the desired locations by selecting the corresponding observer gain K using the eigenvalue assignment technique [14] (traditionally known as the pole-placement technique). If an observer is to be used only for the purpose of observing the system states rather than for feedback control, (9) is less important and the observer closed-loop eigenvalues only need be faster than the fastest system dynamics.

Separation Principle

It is important to point out that the system-observer configuration preserves the closed-loop system eigenvalues that would have been obtained if the linear, perfect-state-feedback control had been used. This fact is shown below. The system (4) under perfect state-feedback control, that is, $u(x(t)) = -Fx(t)$ has the closed-loop form

$$\dot{x}(t) = (A - BF)x(t), \tag{10}$$

so that the eigenvalues of the matrix $A - BF$ are the closed-loop system eigenvalues under perfect state feedback. In the case of the system-observer configuration, the actual control signal applied to both the system and the observer is

$$u(\hat{x}(t)) = -F\hat{x}(t) = -Fx(t) + Fe(t). \tag{11}$$

From (8), (10), and (11),

$$\begin{bmatrix} \dot{x}(t) \\ \dot{e}(t) \end{bmatrix} = \begin{bmatrix} A - BF & BF \\ 0 & A - KC \end{bmatrix} \begin{bmatrix} x(t) \\ e(t) \end{bmatrix} = \mathbf{A} \begin{bmatrix} x(t) \\ e(t) \end{bmatrix}. \tag{12}$$

Since the state matrix of this augmented system is upper block triangular, its eigenvalues are equal to the union of the eigenvalues of the matrices $A - BF$ and $A - KC$. A very simple relation among $x(t)$, $e(t)$, and $\hat{x}(t)$ can be written using the definition of the estimation error as

$$\begin{bmatrix} x(t) \\ e(t) \end{bmatrix} = \begin{bmatrix} I & 0 \\ I & -I \end{bmatrix} \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix} = \mathbf{T} \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix}. \tag{13}$$

Note that the matrix $\mathbf{T}$ is nonsingular. To go from xe-coordinates to $x\hat{x}$-coordinates, the similarity transformation defined in (13) is used, which has a property of preserving the eigenvalues [1]–[9], that is, the eigenvalues $\lambda(A - BF)$and $\lambda(A - KC)$ obtained in the xe-coordinates are identical to the system-observer eigenvalues in $x\hat{x}$-coordinates. The fact that the system-observer configuration has the closed-loop eigenvalues separated into the original system closed-loop eigenvalues (obtained under perfect state feedback) and the actual observer closed-loop eigenvalues is known as the *separation principle*. Hence, the system and the observer eigenvalues can be independently placed (assigned) into desired locations using the system gain F and the observer gain K.

Full-Order Observer Implementation in Simulink

An observer, being an artificial dynamic system of the same order as the original system, can be built by a control engineer using either capacitors and resistors (what electrical engineers do) or using masses, springs, and frictional elements (what mechanical engineers prefer) or simply using a personal computer that simulates and solves the corresponding differential equation, which is something that anybody with a basic knowledge of control systems and differential equations can do, especially those familiar with Matlab and Simulink.

As described in the previous section, an *observer has the same structure as the original system plus the driving feedback term that carries information about the observation error*, that is

$$\begin{aligned} \dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + K(y(t) - \hat{y}(t)), \\ \hat{y}(t) &= C\hat{x}(t). \end{aligned} \tag{14}$$

In practice, the observer is implemented as a linear dynamic system driven by the original system input and output signals, that is, $u(t)$ and $y(t)$. Eliminating $\hat{y}(t)$ from the observer equation (14) yields the observer form used in practical implementations

$$\dot{\hat{x}}(t) = (A - KC)\hat{x}(t) + Bu(t) + Ky(t). \tag{15}$$

The corresponding block diagram of the system-observer configuration, also known as the observer-based controller configuration, is presented in Figure 1 with the feedback loop closed using the feedback controller as defined in (11), namely $u(\hat{x}(t)) = -F\hat{x}(t)$. This block diagram can be implemented using Matlab/Simulink, as presented in Figure 2(a). In Figure 2(a), the rectangular blocks at the end of signal flows (sinks) pass the identified data (variables) to the Matlab workspace. The clock generates the time vector and passes it to the Matlab workspace. The signals may be observed on the scopes or plotted in Matlab using, for example, `plot(t,y)`. This block diagram is edited and presented in Figure 2(b) to improve clarity, as explained in the figure caption.

In Figure 2, the *state-space block* of Simulink is used. That block allows only one input vector and one output vector. For that reason, the observer in (15) is represented as a system with one augmented input

$$\dot{\hat{x}}(t) = (A - KC)\hat{x}(t) + [B \;\; K] \begin{bmatrix} u(t) \\ y(t) \end{bmatrix}. \tag{16}$$

Using the given dimensions of the state, input, and output variables (respectively n, m, and c), the system matrices for the state-space Simulink blocks are set as follows. For the system, the matrices are respectively: `A, B, C, zeros(c,m)`, the last one meaning a zero matrix of dimensions $c \times m$. Note that in the original state-space form, the input signal is not present in the system measurements so that $D = 0$, and this matrix is of dimensions $c \times m$. The system initial condition is unknown, but to run a simulation requires an initial condition (denoted x0). Any vector of dimension n is acceptable. Hence, the system state-space block setup in Simulink is

System State-Space Block Setup: `A, B, C, zeros(c,m), x0`. (17)

After opening the system state-space block in the Simulink window, it is possible to set up the system state-space data.

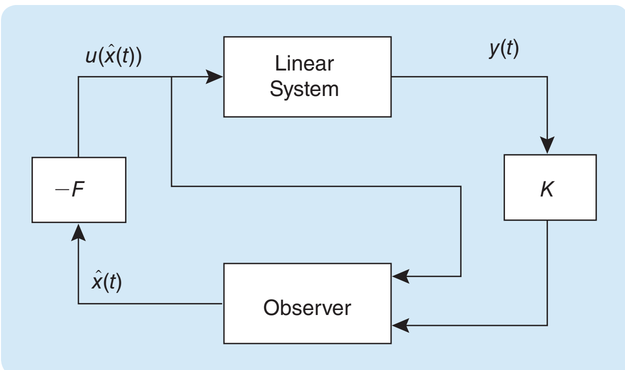

FIGURE 1 A block diagram for the system observer configuration.

Namely, a new window will open (see Figure 3) to input the system data. *These matrices must be previously defined in the Matlab window by running the corresponding Matlab program.*

The results obtained for $y(t)$ and $\hat{y}(t)$, and output observation error $e(t)$, may be presented using the Simulink block "scope," or passed to the Matlab window, together with time, where they can be plotted using the Matlab `plot` function. For example, `plot(t,error)` plots the output observation error as a function of time t. Such a plot (or figure) can be further edited using numerous Matlab figure-editing functions. For the observer, the corresponding matrices should be set using information from (16)

Observer State Space
Block Setup:
`A-K*C,[B K],eye(n),`
`zeros(n,m+c),x0hat.` (18)

The first matrix is the observer feedback matrix, the second is the augmented matrix of two inputs into the observer [see (16)], and the third matrix (set to the identity of dimension n) indicates that estimates of all n state variables are available on the observer output. Since the observer is a system either built by a designer or a computer program that simulates observer dynamics, the designer has full freedom to choose the observer output matrix and set it to an identity matrix so that all observed (estimated) state-space variables appear on the observer output. The fourth matrix represents the matrix "`D`" in the observer block and, due to the fact that the input matrix into the observer is the augmented matrix `[B K]` of dimension $n \times (m+c)$ and the output matrix from the observer is of dimension $n \times n$ (identity matrix I_n), the dimension of the zero matrix must be $n \times (m+c)$. Finally, the last entry in (18) denotes the observer initial condition vector that can be set arbitrarily. Later on in the article, a rational choice of the observer initial condition will be discussed. These matrices can be entered in the Simulink state-space block by double left clicking on the observer state-space block, which will open a new window as shown in Figure 4. For details, see "Matlab Code: System State-Space Block Setup."

The observer initial condition has to be specified. Usually, it

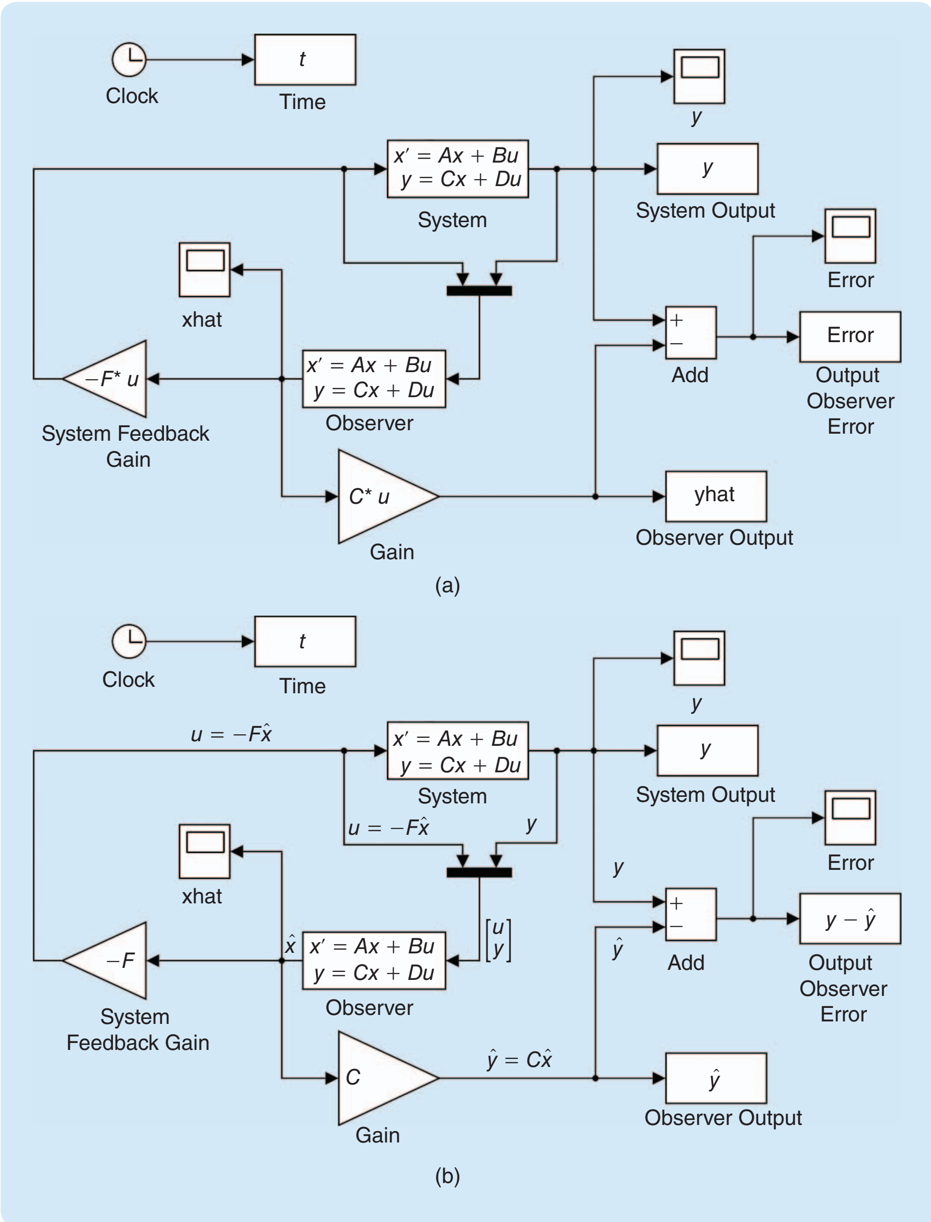

FIGURE 2 (a) A Simulink implementation of the system-observer configuration, the actual block diagram produced by Simulink. Note that the triangular blocks representing a product of a matrix and a vector have in Simulink the "Matrix·vector" notation with the vector being denoted by u. The matrix can be specified and hence changed, but u cannot be changed since it denotes by default all incoming signals. (b) A Simulink implementation of the system-observer configuration, the edited Simulink block diagram, to improve clarity of the Simulink block diagram presented in (a) with labeling the signal wires and to make signal notation consistent with the notation used in the article.

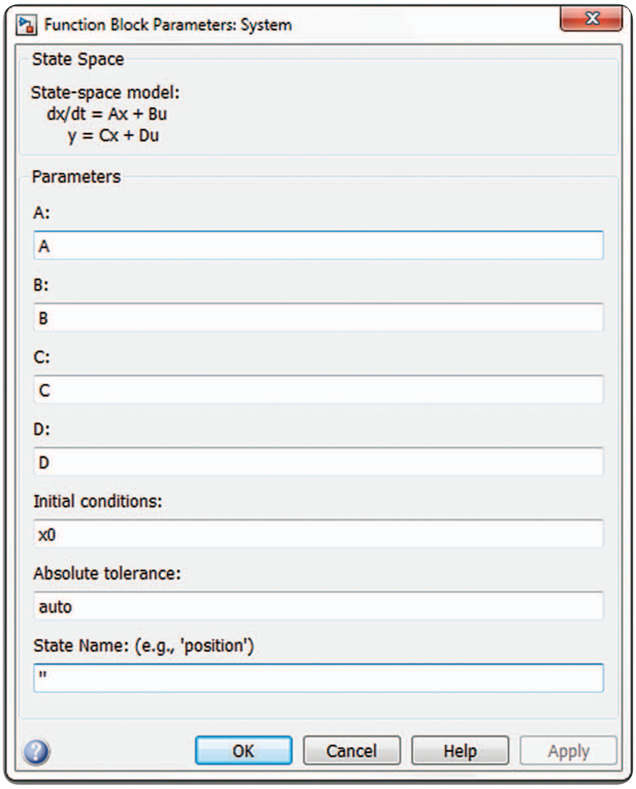

FIGURE 3 The system state-space Simulink data-input window.

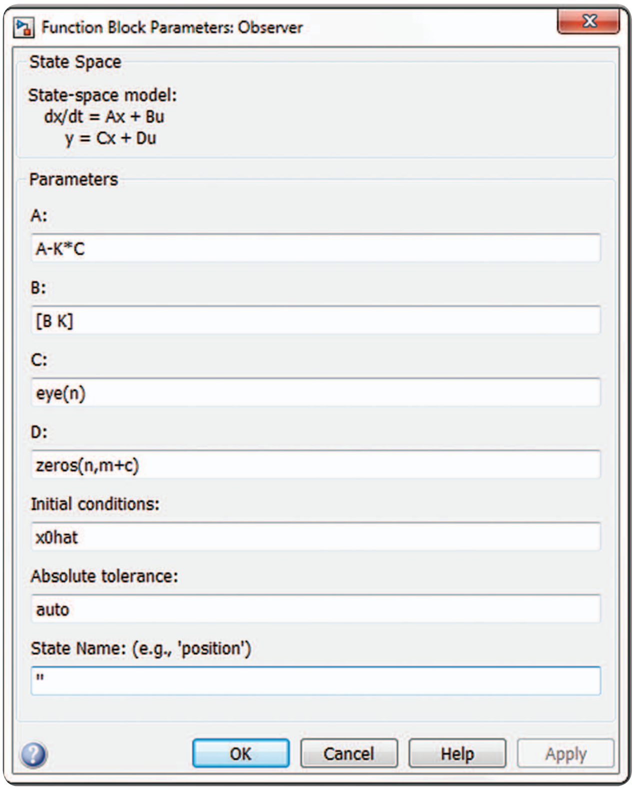

FIGURE 4 The observer state-space Simulink data-input window.

can be any vector whose dimension is equal to the dimension of the system state-space variables. Engineering experience can help one choose the observer's initial condition, but no general guidelines exist. In practice, engineers often set $\hat{x}(0) = 0$ [15] so that $e(0) = x(0) - \hat{x}(0) = x(0)$. Since the matrix C is not, in general, invertible, the initial condition cannot be directly obtained from $y(0) = Cx(0)$. It was recommended in [15] that the partial information about the system initial condition available from the system measurements at the initial time, $y(0) = Cx(0)$, be used to obtain an estimate of $\hat{x}(0)$. In such a case, $\hat{x}(0)$ can be obtained via the least-squares method as

$$\hat{x}(0) = (C^T C)^{-1} C^T y(0), \tag{19}$$

where $y(0)$ is a known measured signal at $t = 0$. Since it was assumed after (3) that matrix C has full rank, that is, $\dim\{y(t)\} = l = c < n = \dim\{x(t)\}$, $c = \mathrm{rank}\{C\}$, the inversion defined in (19) does not exist and a pseudoinverse has to be used [15].

REDUCED-ORDER OBSERVER DESIGN

Consider the linear dynamic system defined in (4) with the corresponding measurements. Assume that the output matrix C has full rank (equal to c) so that there are no redundant measurements. This means that the output equation $y(t) = Cx(t)$ represents, at any time, c linearly independent algebraic equations for n unknown state variables $x(t)$. Note that $y(t)$, of dimension c, is the measured system output, and hence is known at all times. In this section, it is shown how to construct an observer of reduced order $r = n - c$ for estimating the remaining r state-space variables [c state variables can be obtained directly from

Matlab Code: System State-Space Block Setup

```
% System State-Space Block Setup
n = input('dimension of state vector');
    %% dimension of system;
m =input('dimension of input vector');
c = input('dimension of output vector');
% dimension of output vector = l = c = rank(C);
A=A; B=B; C=C; D=zeros(c,m);
  %% assumed D is a zero matrix
% to be able to run simulation we must assign
% some value to the system initial condition
% since in practice this value is given, but
% unknown, that is
x0 = input('some column vector of dimension
   n');
% Observer State-Space Block Setup
Aobs=A-K*C; Bobs=[B K]; Cobs=eye(n);
  Dobs=zeros(n,m+c);
xobs=input('any column vector of dimension
  n')
```

the system measurements $y(t) = Cx(t)$]. As indicated in [3], the reduced-order observer should be used when the system measurements have no noise. If noise is present, it is better to use the full-order observer, since it filters the system measurements and, in general, all state variables.

It will be seen that the procedure for obtaining the reduced-order observer is not unique. An arbitrary matrix C_1 of dimension $r \times n$ whose rank is equal to $r = n - c$ can be found such that the augmented matrix

$$\text{rank}\begin{bmatrix} C \\ C_1 \end{bmatrix} = n, \tag{20}$$

has full rank equal to n. Introduce a vector $p(t)$ of dimension r such that

$$p(t) = C_1 x(t). \tag{21}$$

Combining (3) and (21) yields

$$\begin{bmatrix} y(t) \\ p(t) \end{bmatrix} = \begin{bmatrix} C \\ C_1 \end{bmatrix} x(t),$$

which can be solved to obtain

$$x(t) = \begin{bmatrix} C \\ C_1 \end{bmatrix}^{-1} \begin{bmatrix} y(t) \\ p(t) \end{bmatrix} = [L \;\; L_1] \begin{bmatrix} y(t) \\ p(t) \end{bmatrix} = Ly(t) + L_1 p(t). \tag{22}$$

An estimate of $x(t)$ can be obtained from (22) as

$$\hat{x}(t) = Ly(t) + L_1 \hat{p}(t), \tag{23}$$

where $y(t)$ are known system measurements and $\hat{p}(t)$ has to be obtained using a reduced-order observer of dimension $r < n$. As a preliminary result in the process of constructing a reduced-order observer for $p(t)$, it can be noticed from (22) that the following algebraic relations exist

$$\begin{bmatrix} C \\ C_1 \end{bmatrix} \begin{bmatrix} C \\ C_1 \end{bmatrix}^{-1} = I_n = \begin{bmatrix} C \\ C_1 \end{bmatrix} [L \;\; L_1] = \begin{bmatrix} CL & CL_1 \\ C_1 L & C_1 L_1 \end{bmatrix} = \begin{bmatrix} I_c & 0 \\ 0 & I_r \end{bmatrix} \tag{24}$$

giving $CL = I_c$, $C_1 L_1 = I_r$, $CL_1 = 0$, $C_1 L = 0$, where I_n, I_c, I_r, are identity matrices of corresponding dimensions and $n = c + r$.

Since from (21) $p(t) = C_1 x(t)$, the differential equation for $p(t)$ can be easily constructed using the original system differential equation (4) and (22), which leads to

$$\begin{aligned} \dot{p}(t) &= \frac{d}{dt}\{p(t)\} = \frac{d}{dt}\{C_1 x(t)\} = C_1 \dot{x}(t) \\ &= C_1 A x(t) + C_1 B u(t) = C_1 A L_1 p(t) + C_1 B u(t). \end{aligned} \tag{25}$$

Trying to design an observer for $p(t)$ using the above-established principle [see (14); an *observer has the same structure as the original system plus the driving feedback term that carries information about the observation error*], it can be found that $y(t)$ does not contain information about $p(t)$ since from (22) and (24)

$$y(t) = Cx(t) = CLy(t) + CL_1 p(t) = y(t) + 0 = y(t). \tag{26}$$

More information about $y(t)$ can be obtained from the knowledge of $\dot{y}(t)$, in which case

$$\begin{aligned} \dot{y}(t) &= C\dot{x}(t) = CAx(t) + CBu(t) \\ &= CAL_1 p(t) + CALy(t) + CBu(t). \end{aligned} \tag{27}$$

This indicates that $\dot{y}(t)$ contains information about $p(t)$, so that it can be used to construct an observer for $p(t)$

$$\dot{\hat{p}}(t) = C_1 A L_1 \hat{p}(t) + C_1 A L y(t) + C_1 B u(t) + K_1(\dot{y}(t) - \dot{\hat{y}}(t)), \tag{28}$$

where K_1 is the reduced-order observer gain, which has to be determined such that the reduced-order observer error $p(t) - \hat{p}(t)$ goes to zero as time increases. The reduced-order observer measurement is obtained from (27) by replacing $p(t)$ by its estimate (observation) $\hat{p}(t)$, that is,

$$\dot{\hat{y}}(t) = CAL_1 \hat{p}(t) + CALy(t) + CBu(t). \tag{29}$$

Signal differentiation is not a recommended operation since it is very sensitive to noise; signal differentiation amplifies noise so it should be avoided in all practical applications. Using an appropriate change of variables, it is possible to completely eliminate the need for information about the derivative of the system measurements $\dot{y}(t)$.

Eliminating $\dot{\hat{y}}(t)$ from (28) using (29), produces

$$\begin{aligned} \dot{\hat{p}}(t) &= C_1 A L_1 \hat{p}(t) + C_1 A L y(t) + C_1 B u(t) \\ &\quad + K_1(\dot{y}(t) - CAL_1 \hat{p}(t) + CALy(t) + CBu(t)). \end{aligned} \tag{30}$$

Bringing the term with $\dot{y}(t)$ to the left-hand side of (30) and introducing a change of variables

$$\hat{q}(t) = \hat{p}(t) - K_1 y(t) \text{ which implies } \dot{\hat{q}}(t) = \dot{\hat{p}}(t) - K_1 \dot{y}(t), \tag{31}$$

the reduced-order observer for $\hat{q}(t)$ is

$$\dot{\hat{q}}(t) = A_q \hat{q}(t) + B_q u(t) + K_q y(t), \tag{32}$$

where

$$\begin{aligned} &A_q = (C_1 - K_1 C) A L_1, \; B_q = (C_1 - K_1 C) B, \\ &K_q = (C_1 - K_1 C) A (L + L_1 K_1). \end{aligned} \tag{33}$$

Using the observation for $\hat{q}(t)$ obtained from (32)–(33), the reduced-order observer estimate is obtained from (31) as $\hat{p}(t) = \hat{q}(t) + K_1 y(t)$, which gives the estimate of the original state-space variables as

$$\begin{aligned} \hat{x}(t) &= Ly(t) + L_1 \hat{p}(t) = Ly(t) + L_1(\hat{q}(t) + K_1 y(t)) \\ &= (L + L_1 K_1) y(t) + L_1 \hat{q}(t). \end{aligned} \tag{34}$$

In Figure 5, the obtained system-reduced-observer configuration block diagram produced by Simulink is presented in (a), and this block diagram is edited and presented in (b), where signal wires are labeled and the reduced-order observer equation (32) and its Simulink implementation are denoted. As shown in Figure 6, for the reduced-order observer, the corresponding matrices are set as

Observer State-Space Block Setup: `Aq, [Bq  Kq],`
`eye(n-c), zeros(n-c,m+c), x0hat_reduced.` (35)

Comment

An interesting observation can be made about the signal $\hat{y}(t) = C\hat{x}(t)$, namely, using the result established in (24)

$$\begin{aligned}\hat{y}(t) &= C\hat{x}(t)\\ &= C(L_1\hat{q}(t)\\ &\quad + (L + L_1K_1)y(t))\\ &= CL_1\hat{q}(t) + CLy(t)\\ &\quad + CL_1K_1y(t)\\ &= 0\hat{q}(t) + Iy(t)\\ &\quad + 0y(t)\\ &= y(t)\end{aligned} \tag{36}$$

indicating that the reduced-order observer also produces in its output information about the exact values of the system actual output at any time, that is

$$\begin{aligned}\hat{y}(t) &= C\hat{x}(t)\\ &= y(t), \text{ for all } t \geq t_0 = 0.\end{aligned} \tag{37}$$

This appears to be a new result not previously observed in engineering literature. Hence, the system output and full-order observer output can be compared as in the block diagram in Figure 2(a) and (b) to define the output-observation error signal. However, in the case of the system reduced-order configuration $y(t) - \hat{y}(t)$ produces signal equal to zero at all times.

Like in the case of the full-order observer design, the designer is faced with two fundamental problems: how to choose the reduced-order observer initial conditions and how to find the reduced-order observer gain that stabilizes the reduced-order observer and sets its eigenvalues in the desired locations (to make the reduced-order observer much faster than the system). These questions will be answered in the following subsections.

Least-Squares Choice for the Reduced-Order Observer Initial Conditions

The least-squares solution for the choice of the reduced-order observer initial conditions can be obtained by extending the least-squares solution of the full-order observer

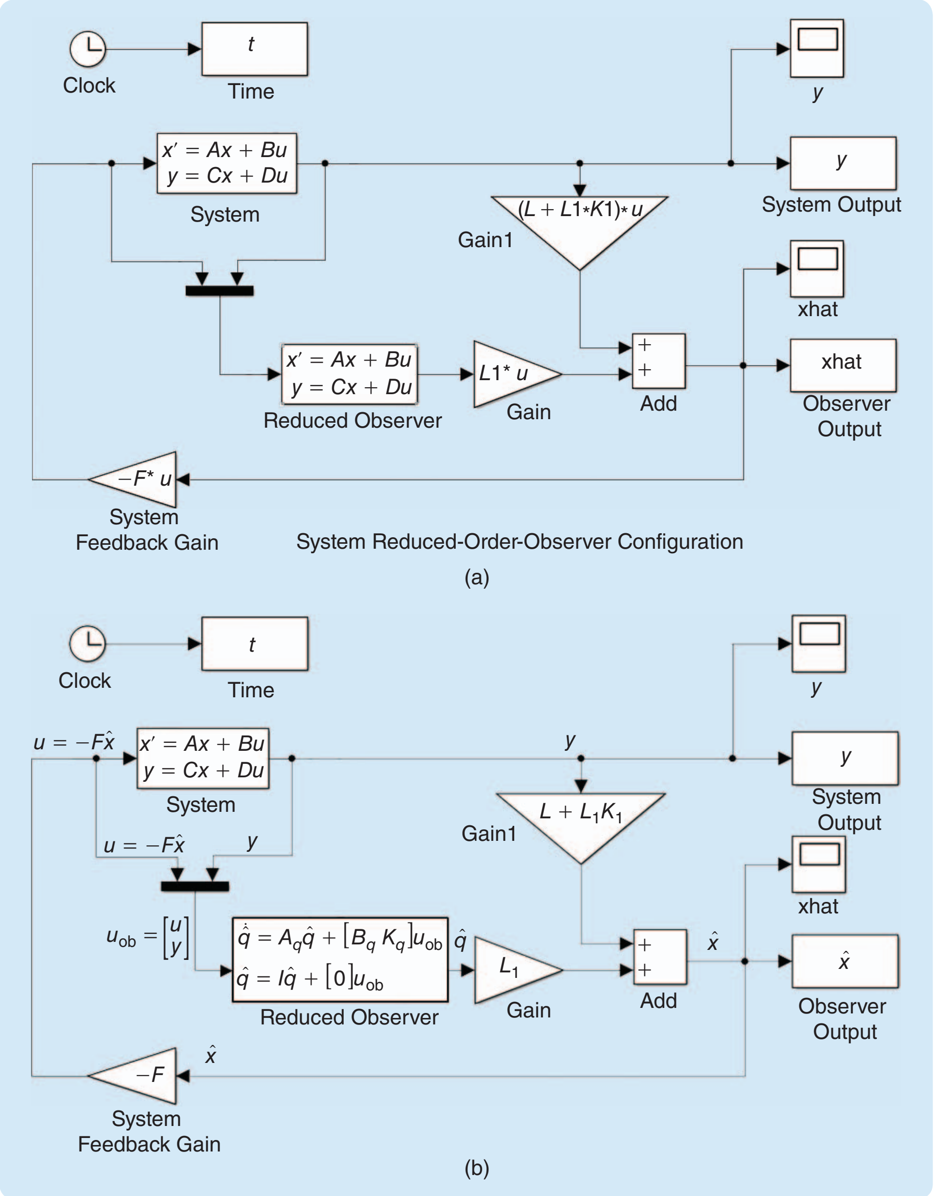

FIGURE 5 (a) A Simulink implementation of the system-observer configuration, the actual block diagram produced by Simulink. Note that the triangular blocks representing a product of a matrix and a vector have in Simulink the "Matrix*vector" notation with the vector being denoted by *u*. The matrix can be specified and hence changed, but *u* cannot be changed since it denotes by default all incoming signals. (b) A Simulink implementation of the system-observer configuration, the edited Simulink block diagram, to improve clarity of the Simulink block diagram presented in (a) with labeling the signal wires and to make signal notation consistent with the notation used in the article.

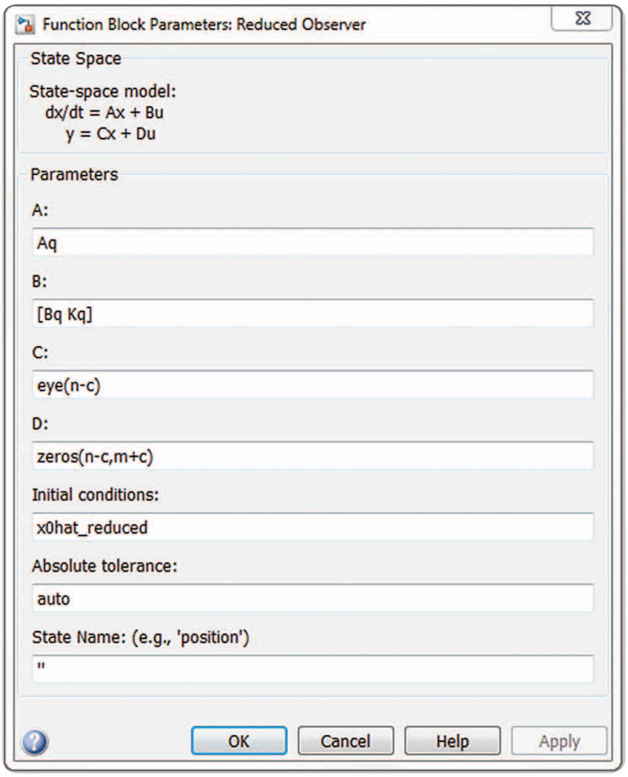

FIGURE 6 A reduced-order observer state-space Simulink data-input window.

Reduced-Order-Observer Design Process

Step 1: Check the observability condition.

Step 2: Find the closed-loop system eigenvalues.

Step 3: Select the closed-loop reduced-order observer eigenvalues

Step 4: Find the reduced-order observer feedback gain.

The corresponding Matlab code is given by:

```
% Step 1:
O=obsv(C1*A*L1,C*A*L1); rank(O);
 %% rank must be equal to r=n-c
% Step 2:
lambda_sys=eig(A-B*F);
% Step 3: input desired lambda_obs (reduced-
% order observer eigenvalues)
lambda_obs=input('input desired observer
eigenvalues')
% Step 4:
K1T=place((C1*A*L1)', (C*A*L1)', lambda_
 obs); K1=K1T'
```

initial conditions of [15] [given in (19)] to the reduced-order observer case. Equation (34) at the initial time $t = 0$ produces

$$\delta\hat{x}(0) = L_1\delta\hat{q}(0) + (L + L_1K_1)\delta y(0). \tag{38}$$

Since the matrix C has full rank equal to c, the least-squares solution for $\hat{q}(t)$ is

$$\begin{aligned} L_1\delta\hat{q}(0) &= \delta\hat{x}(0) - (L + L_1K_1)\delta y(0) \\ &= [(C^TC)^{-1}C^T - (L + L_1K_1)]\delta y(0), \end{aligned} \tag{39}$$

which produces

$$\delta\hat{q}(0) = (L_1^TL_1)^{-1}L_1^T[(C^TC)^{-1}C^T - (L + L_1K_1)]\delta y(0). \tag{40}$$

It follows from (24) that L_1 has full rank equal to r, $C_1L_1 = I_r$, $r = n - c$ so that the corresponding inversion $L_1^TL_1$ exists. Equation (40) presents a novel theoretical result regarding setting the *reduced-order observer* initial condition using the least-squares method.

Setting Reduced-Order-Observer Eigenvalues in the Desired Locations

As discussed in (9), good performance typically requires placing the reduced-order observer eigenvalues such that the reduced-order observer is roughly ten times faster than the system whose speed is determined by the closed-loop system eigenvalues given by $\lambda(A - BF)$. Note that the reduced-order observer matrix can be written as

$$A_q = C_1AL_1 - K_1CAL_1 = (C_1AL_1) - K_1(CAL_1). \tag{41}$$

To determine the reduced-order observer gain K_1 such that it arbitrarily places the reduced-order observer eigenvalues, the pair (C_1AL_1, CAL_1) must be observable. The next subsection will show that this condition is satisfied if the original system is observable (the pair (A, C) is observable). See "Reduced-Order-Observer Design Process" for details on the steps involved in the design of this observer.

Reduced-Order Observer Design with a Change of State Coordinates

It is shown first in this subsection that a reduced-order observer can be also designed with a change of state coordinates. More importantly, with this change of state coordinates, it becomes easy to prove the result that if the original system is observable then the reduced-order observer is also observable.

Consider a linear system with the corresponding measurements defined in (4). The reduced-order observer design can be simplified, and some important conclusions can easily be made using a nonsingular transformation $P^{n\times n}$

$$Px(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \tag{42}$$

such that

$$\begin{aligned} y(t) = Cx(t) = CP^{-1}Px(t) &= CP^{-1}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \\ &= [I_c \;\; 0]\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = x_1(t), \end{aligned} \tag{43}$$

to map the system into new coordinates in which the *system measurements provide complete information about a part of the state-space vector* $x_1(t)$ of dimension c. In such a case, an observer is needed to estimate the remaining part of the state-space vector, that is, $x_2(t)$ of dimension $r = n - c$. This is possible since there exists a transformation P of the full-rank matrix C, obtained using elementary transformations, such that $CP^{-1} = [I_c \ \ 0]$. This procedure can be found in many standard linear algebra textbooks, for example [16]. Such a transformation is defined in (24) with $P^{-1} = [L \ \ L_1]$ so that $CP^{-1} = C[L \ \ L_1] = [CL \ \ CL_1] = [I \ \ 0]$.

In the new coordinates, the state transformation (42) produces

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(t),$$
$$x_2(t_0) = x_{20} = \text{unknown},$$
$$y(t) = x_1(t). \tag{44}$$

Since $x_1(t)$ is directly measured at all times, its estimate is $\hat{x}_1(t) = y(t)$. An observer for $x_2(t)$ can be constructed in a standard manner using the observer design principle: *an observer has the same structure as the system plus a driving feedback term coming from the system measurements that carries information about the observation error*, that is

$$\dot{\hat{x}}_2(t) = A_{21}x_1(t) + A_{22}\hat{x}_2(t) + B_2u(t) + K_2(y(t) - \hat{y}(t)). \tag{45}$$

But this design does not produce information about $e_2(t) = x_2(t) - \hat{x}_2(t)$ since $y(t)$ provides complete information about $x_1(t)$ and no information about $x_2(t)$. To obtain more information, it is necessary to know the derivative of $y(t)$. This derivative is

$$\dot{y}(t) = \dot{x}_1(t) = A_{11}x_1(t) + A_{12}x_2(t) + B_1u(t), \tag{46}$$

so that the reduced-order observer measurements are

$$\dot{\hat{y}}(t) = A_{11}x_1(t) + A_{12}\hat{x}_2(t) + B_1u(t), \tag{47}$$

and the reduced-order observer becomes

$$\dot{\hat{x}}_2(t) = A_{21}x_1(t) + A_{22}\hat{x}_2(t) + B_2u(t) + K_2(\dot{y}(t) - \dot{\hat{y}}(t)),$$
$$\dot{\hat{y}}(t) = A_{11}y(t) + A_{12}\hat{x}_2(t) + B_1u(t). \tag{48}$$

The need for $\dot{y}(t)$ can be eliminated by introducing a new variable

$$\hat{z}_2(t) = \hat{x}_2(t) - K_2y(t), \tag{49}$$

which produces the final form for the reduced-order observer

$$\dot{\hat{z}}_2(t) = A_z\hat{z}_2(t) + B_zu(t) + K_zy(t), \tag{50}$$

where

$$A_z = A_{22} - K_2A_{12},$$
$$B_z = B_2 - K_2B_1,$$
$$K_z = A_{21} - K_2A_{11} + (A_{22} - K_2A_{12})K_2. \tag{51}$$

The expression for dynamics of the estimation error can be obtained from the observation error definition, that is, $\dot{e}_2(t) = \dot{x}_2(t) - \dot{\hat{x}}_2(t)$, leading to

$$\dot{e}_2(t) = (A_{22} - K_2A_{12})e_2(t), \tag{52}$$

which indicates that the estimation error $e_2(t)$ decays to zero if $A_{22} - K_2A_{12}$ is an asymptotically stable matrix. The eigenvalues of matrix $A_{22} - K_2A_{12}$ can be arbitrarily located via the feedback matrix K_2 if the pair (A_{22}^T, A_{12}^T) is controllable, which by duality between controllability and observability is equivalent to the requirement that the pair (A_{22}, A_{12}) is observable. Below it is shown that having the pair (A, C) observable implies that the pair (A_{22}, A_{12}) is also observable.

Using the Popov–Belevitch–Hautus observability test [14], the pair (A, C) is observable if the following matrix has rank n for all eigenvalues of matrix A

$$\text{rank}\left\{\begin{bmatrix} A - \lambda_i I \\ C \end{bmatrix}\right\} = n, \quad \text{for all} \quad \lambda_i(A), \quad i = 1, 2, \ldots, n. \tag{53}$$

Using the specific partitioning of matrices A and C as given in (43) and (44), it follows that

$$\text{rank}\left\{\begin{bmatrix} A_{11} - \lambda_i I_c & A_{12} \\ A_{21} & A_{22} - \lambda_i I_r \\ I_c & 0 \end{bmatrix}\right\} = n,$$
$$\text{for all} \quad \lambda_i(A), \quad i = 1, 2, \ldots, n, \tag{54}$$

which is due to the facts that the last c rows have rank c and $n = c + r$. Equation (54) implies the following condition

$$\text{rank}\left\{\begin{bmatrix} A_{22} - \lambda_i I_r \\ A_{12} \end{bmatrix}\right\} = r, \quad \text{for all} \quad \lambda_i(A_{22}), \quad i = 1, 2, \ldots, r, \tag{55}$$

so that the pair (A_{22}, A_{12}) is observable by the direct application of the Popov–Belevitch–Hautus observability test.

Hence, under the condition that the original full-order system is observable, the reduced-order observer can be constructed. The reduced-order observer gain should be chosen such that the reduced-order observer eigenvalues are placed far to the left, which causes that the estimation error to quickly decay to zero.

Extension to Kalman Filtering

By mastering the design of deterministic observers presented in this article, more complex linear dynamic estimation problems can also be considered, for example, the Kalman filtering and other optimal linear estimation problems [17]–[20]. In fact, the Kalman filter can also be implemented as a software program in Matlab/Simulink using the methodology presented in this article. For a linear dynamic system disturbed by a Gaussian white noise stochastic process $w(t)$ and system measurements corrupted by a Gaussian white noise stochastic process $v(t)$.

$$\dot{x}(t) = Ax(t) + Bu(t) + Gw(t),$$
$$y(t) = Cx(t) + v(t), \tag{56}$$

the Kalman filter is given by [17]–[20]

$$\dot{\hat{x}}_{KF}(t) = (A - KC)\hat{x}_{KF}(t) + Bu(t) + K_{KF}y(t), \tag{57}$$

which is exactly the same structure as the full-order observer structure presented in (15). The Kalman filter gain K_{KF} and the Kalman filter initial conditions must be chosen using results from optimal Kalman filtering theory.

FULL- AND REDUCED-ORDER OBSERVER DESIGNS FOR AN AIRCRAFT MODEL

The state-space matrices for an aircraft model are [7]

$$A = \begin{bmatrix} -0.01357 & -32.2 & -46.3 & 0 \\ 0.00012 & 0 & 1.214 & 0 \\ -0.0001212 & 0 & -1.214 & 1 \\ 0.00057 & 0 & -9.1 & -0.6696 \end{bmatrix},$$
$$B = \begin{bmatrix} -0.433 \\ 0.1394 \\ -0.1394 \\ -0.1577 \end{bmatrix},$$
$$C = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$
$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Matrix C_1 needed for the reduced-order observer design is chosen as

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

See "Matlab Program for the Aircraft Simulation Example" for the corresponding Matlab program with all simulation data. The obtained differences between the actual and estimated state trajectories using the full- and reduced-order observers are presented in Figures 7 and 8. In both cases, the initial conditions for the observers are obtained using the least-squares formulas (19) and (40). It can be seen from Figures 7 and 8 that the reduced-order observer performs better than the full-order observer. Not only are there no observation errors for the two state variables directly measured, but the reduced-order observer is also more accurate than the full-order observer. Note that the eigenvalues for both observers are placed to be of the same speed. The reduced-order observer eigenvalues are placed at −10, −11 and the full-order observer eigenvalues are placed at −10, −11, −12, −13. Moreover, the reduced-order observer is simpler for implementation, since it is a dynamic system of lower order than the original system. An additional advantage of the reduced-order observer is that they require fewer sensors and fewer feedback loops for corresponding feedback control applications.

The importance of Matlab/Simulink implementation of the full- and reduced-order observers has been demonstrated in the previous sections of this article. These results can be used in all areas of science and engineering. For example, Matlab/Simulink simulation and implementation of switching converters and power converters is nicely demonstrated in [21]–[22].

Matlab Program for the Aircraft Simulation Example

```
% n=system order; m=number inputs;
% l=number of outputs; c=rank of the output
  matrix
n=4; m=1; l=2; c=2;
A=[-0.01357 -32.2 -46.3 0;
    0.00012 0 1.214 0;
    -0.0001212 0 -1.214 1;
    0.00057 0 -9.1 -0.6696]
B=[-0.433;0.1394;-0.1394;-0.1577]
C=[0 0 0 1;
   1 0 0 0];
D=[0;0]
x0=[2;2;2;2];
lambdaSystemDesired=[-0.5, -1+j*1, -1-j*1, -2]
F=place(A,B,lambdaSystemDesired)
lambdaObsDesired=[-10 -11 -12 -13];
KT=place(A',C',lambdaObsDesired);
K=KT'
% Reduced-order Observer Design
C1=[0 1 0 0;0 0 1 0];
Caug=[C;C1]; Laug=inv(Caug);
L=Laug(1:n,1:c); L1=Laug(1:n,c+1:n)
lambda_red_obs=[-10 -11];
K1T=place((C1*A*L1)',(C*A*L1)',lambda_red_
  obs);
K1=K1T'
Aq=C1*A*L1-K1*C*A*L1;
Bq=C1*B-K1*C*B;
Kq=C1*A*L1*K1+C1*A*L-K1*C*A*L-K1*C*A*L1*K1;
% Least-Squares Choices for Initial Conditions
y0=C*x0;
x0hat=pinv(C'*C)*C'*y0;
x0hat_reduced=inv(L1'*L1)*L1'*(pinv(C'*C)
  *C'-(L+L1*K1))*y0;
```

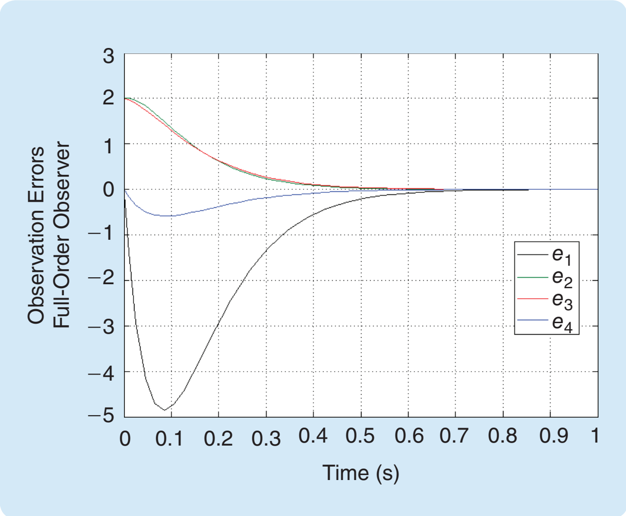

FIGURE 7 Full-order observer estimation errors $e(t) = x(t) - \hat{x}(t)$ for the aircraft example.

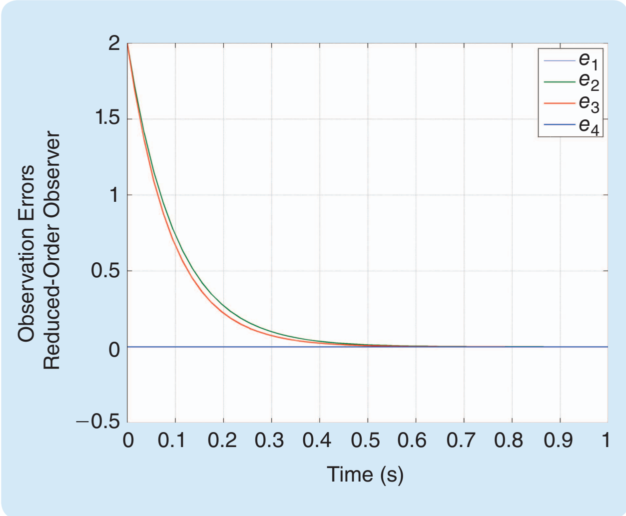

FIGURE 8 Reduced-order observer estimation errors $x(t) - \hat{x}(t)$ with $\hat{x}(t)$ obtained from (34) for the aircraft example.

CONCLUSIONS

In this tutorial, It has been shown in detail how to implement full- and reduced-order observers in Matlab/Simulink. The corresponding fundamental derivations and results have been presented. This presentation can inspire undergraduate and graduate students to further study observers and use them as powerful tools for observing system dynamics and/or designing feedback control loops. Hopefully, control engineers will use observers more often in practice because they are easily implemented on personal computers.

AUTHOR INFORMATION

Verica Radisavljevic-Gajic (verica.gajic@villanova.edu) received her doctoral degree from Rutgers University, Department of Mechanical and Aerospace Engineering, under the direction of Prof. Haim Baruh in modeling and control of complex dynamic systems described by differential-algebraic equations. She held visiting professor positions at Rutgers University and Lafayette College and a postdoctoral position in systems biology. From 2009 to 2012 she was an assistant professor of mechanical engineering at California State University, Los Angeles (she was on leave at the American University of Sharjah, United Arab Emirates from 2010 to 2012). Since 2012 she has been Clare Boothe Luce Assistant Professor of Mechanical Engineering at Villanova University. She has four years of industrial experience in control systems. Her research interests are in modeling and control of complex dynamic mechanical systems and systems in biology and medicine. One of the main areas of her research focuses on control of distributed-parameter systems (with applications to heat and beam equations). Recently, she has become interested in power and energy systems, particularly in dynamic models and control of proton exchange membrane fuel cells. She has published over 20 journal papers in the leading journals in these fields.

REFERENCES

[1] N. Nise, *Control Systems Engineering*. Hoboken, NJ: Wiley, 2008.
[2] R. Stefani, C. Savant, B. Shahian, and G. Hostetter, *Design of Feedback Control Systems*. Orlando, FL: Saunders College Publishing, 1994.
[3] G. Franklin, J. Powel, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Upper Saddle River, NJ: Prentice Hall, 2002.
[4] R. Dorf and R. Bishop, *Modern Control Systems*. Upper Saddle River, NJ: Pearson Education, 2005.
[5] K. Ogata, *Modern Control Engineering*. Upper Saddle River, NJ: Prentice Hall, 2002.
[6] S. Shinners, *Modern Control Systems Theory and Design*. New York: Wiley, 1992.
[7] Z. Gajic and M. Lelic, *Modern Control Systems Engineering*, London, U.K.: Prentice Hall, 1996.
[8] B. Friedland, *Control Systems Design: An Introduction to State Space Methods*. Mineola, NY: Dover Publications, 2005.
[9] F. Fairman, "A pedagogical approach to minimal order observers," *IEEE Trans. Educ.*, vol. 33, no. 3, pp. 268–269, 1990.
[10] Z. Gajic, *Linear Dynamic Systems and Signals*. Upper Saddle River, NJ: Prentice Hall, 2003.
[11] D. Luenberger, "Observing the state of a linear system," *IEEE Trans. Military Electron.*, vol. 8, no. 2, pp. 74–80, 1964.
[12] D. Luenberger, "Observers for multivariable systems," *IEEE Trans. Autom. Contr.*, vol. 11, no. 2, pp. 190–197, 1966.
[13] D. Luenberger, "An introduction to observers," *IEEE Trans. Autom. Contr.*, vol. 16, no. 6, pp. 596–602, 1971.
[14] C.-T. Chen, *Linear Systems: Analysis and Design*. Oxford, U.K.: Oxford Univ. Press, 2012.
[15] C. D. Johnson, "Optimal initial condition for full-order observers," *Int. J. Control*, vol. 48, no. 3, pp. 857–864, 1988.
[16] G. Strang, *Introduction to Linear Algebra*. Wellesley, MA: Wellesley Cambridge Press, 2009.
[17] A. Gelb, *Applied Optimal Estimation*. Cambridge, MA: MIT Press, 1974.
[18] D. Simon, *Optimal State Estimation, Kalman, H Infinity, and Nonlinear Approaches*. New York: Wiley-Interscience, 2006.
[19] A. Sinha, *Linear System: Optimal and Robust Control*. Boca Raton, FL: CRC Press Taylor and Francis Group, 2007.
[20] M. Grewal and A. Andrews, *Kalman Filtering: Theory and Practice with MATLAB*. New York: Wiley-IEEE Press, 2014.
[21] J.-H. Su, J.-J. Chen, and D.-S. Wu, "Learning feedback controller design of switching converters via MATLAB/SIMULINK," *IEEE Trans. Educ.*, vol. 45, no .4, pp. 307–315, 2002.
[22] V. Peres and J. Silva, "Teaching nonlinear modeling, simulation, and control of electronic power converters using MATLAB/SIMULINK," *IEEE Trans. Educ.*, vol. 45, no. 3, pp. 253–261, 2002.