Linear Observers Design and Implementation

Verica Radisavljevic-Gajic, *Member, IEEE*

Abstract— **In this paper we show how to implement full- and reduced-order observers using the software packages for computer aided control system design MATLAB and SIMULINK. As a matter of fact, we show how to implement a linear system and its observer, represented by their state space forms, using the SIMULINK state space blocks. We discuss how to choose the observer(s) initial conditions and how to setup the observer gain(s). The presentation is done at the level of undergraduate students who have completed a basic undergraduate class in control systems and have knowledge of the state space linear system representation. Having full understanding of the observer implementation(s), students and engineers will feel confident to use these observers and observer based controllers in numerous engineering and scientific applications.**

Index Terms—**Observer Design, Reduced-Order Observer Design, Implementation in MATLAB and SIMULINK.**

I. Introduction

THE design of observers is usually considered as a graduate level topic and taught in a graduate level control engineering course. However, in the most recent editions of several standard undergraduate control system textbooks we can find the coverage of full-order and even reduced-order observers [1]-[8]. In this paper, we clarify how to implement these observers using the software packages for computer aided control system design MATAB and SIMULINK. We discuss also how to choose the observer initial conditions and how to set up the observer gain(s). Note that observers are used these days not only for the purpose of feedback control, but also in their own right to observe state variables of a dynamic system, which can be an experiment in progress whose state has to be monitored (observed, estimated) at all times.

In undergraduate classes on control systems, we deal with time invariant linear systems 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 (that may be used as a system control input) of dimension m, and matrices A and B are constant and of appropriate dimensions. To take the advantage of good features of feedback (see for example Chapter 12 of [9]), it is often assumed that all state variables are available for feedback (full-state feedback), meaning that a feedback control input can be constructed as

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

where F is a constant feedback matrix of dimension $m \times n$. There is an important implementational difficulty of full-state feedback controllers. This difficulty is in the fact that *all state space variables must be available for feedback*, which in the case of higher-order dimensional systems creates a lot of feedback loops and might be very costly and/or not practical. Even more, often some state variables are not available for feedback. Instead, an output signal that represents a linear combination of the state space variables is available only as

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

In general, the dimension of the output signal is much smaller than the dimension of the state space variables, that is, $\dim\{y(t)\} = l = c < n = \dim\{x(t)\}$, where $c = \operatorname{rank}\{C\}$. Hence, we have also assumed that there are not redundant measurements. In such a case, under certain conditions, we can use an *observer*, a dynamic system driven by the system input and output signals with the goal to reconstructing (observing, estimating) at all times all the system state space variables.

II. Full-Order Observer Design

The theory of observers originated in the work of Luenberger in the middle of the 1960s, [10]-[12]. According to Luenberger, *any system driven by the output of the given system can serve as an observer for that system.* Consider a linear dynamic system

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

The system output variables are available at all times, and that information can be used to construct an artificial dynamic system of the same order as the system under consideration, order *n*, which will 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

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

If we compare the outputs $y(t)$ and $\hat{y}(t)$, in general, they will be different since in the first case the initial condition of (4) is unknown, and the initial condition of the proposed observer (5) can be chosen arbitrarily by a control engineer.

Manuscript received February 5, 2014. This work was supported in part by the Clare Boothe Luce Foundation.

V. Radisavljevic-Gajic is with the Mechanical Engineering Department, Villanova University, Villanova, PA 19085 USA (phone: 610-519-6632; fax: 610-519-7312; e-mail: verica.gajic@villanova.edu).

978-1-4799-5233-5/14/$31.00 ©2014 IEEE

The difference between these two outputs defines 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, theoretically goes to zero at steady state. The observer that takes into the account feedback information about the observation error is given by [10]-[12]

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

where the matrix K represents the observer gain that has to be chosen such that the observation error tends to zero as time increases. From (4) and (7), we get an expression for dynamics of the observation error as

$$\dot{e}(t)=(A-KC)e(t), \quad e(t_0)=\text{unknown} \tag{8}$$

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*. By taking the transpose of the estimation error feedback matrix, that is, $A^T-C^TK^T$, it can be seen that the pair (A^T,C^T) must be controllable [1]-[8], in order to place the observer eigenvalues in the left half of the complex plane, and make it asymptotically stable. It is known that controllability of the pair (A^T,C^T) is equal to *observability of the pair* (A,C), which *is the condition needed for the observer design*, [1]-[8]

The observer eigenvalues should be chosen to be about ten times faster than the system eigenvalues. This can be achieved by setting the smallest real part of the observer eigenvalues to be ten times bigger than the largest real part of the closed-loop system eigenvalues

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

Theoretically, an observer can be made arbitrarily fast by pushing its closed-loop eigenvalues very far to the left in the complex plane, but very fast observers generate noise, and this is not desirable. In control system practice, instead of ten times, it is sufficient that the closed-loop observer eigenvalues are faster than the closed-loop system eigenvalues 5-6 times. If an observer is used only for the purpose of observing the system states, the condition (9) is not relevant and the observer closed-loop eigenvalues should be made sufficiently fast (faster than the system dynamics).

A. 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 will be shown in the follow-up of this section. The system (4) under the perfect state feedback control, that is, $u(t)=-Fx(t)$ 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 applied to both the system and the observer is given by

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

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

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

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

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

Note that the matrix T is nonsingular. In order to go from xe-coordinates to $x\hat{x}$-coordinates we use the similarity transformation defined in (13), which has a property of preserving the eigenvalues, [1]-[8], that is, the eigenvalues $\lambda(A-BF)$ and $\lambda(A-KC)$ obtained in the *xe*-coordinates are identical to the system-observer eigenvalues in the $x\hat{x}$-coordinates. This 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.

B. 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 would do) or using masses, springs, and frictional elements (what mechanical engineers would prefer) or using simply a personal computer that simulates and solves the corresponding differential equation what anybody with the basic knowledge of control systems and differential equations can do, especially those familiar with MATLAB and SIMULINK.

As constructed 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

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

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 $y(t)$. Namely, eliminating $\hat{y}(t)$ from the observer equation (7), we obtain the following 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 $u(t) = -F\hat{x}(t)$. This block diagram can be implemented using MATLAB/SIMULINK as presented in Figure 2.

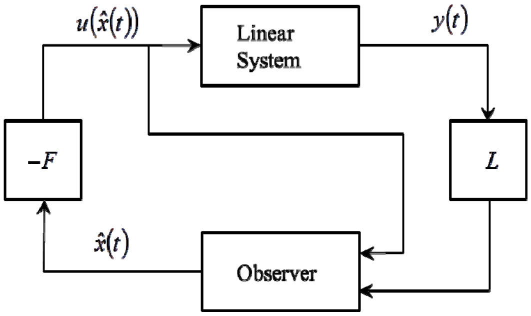

Figure 1: Block diagram for the system observer configuration

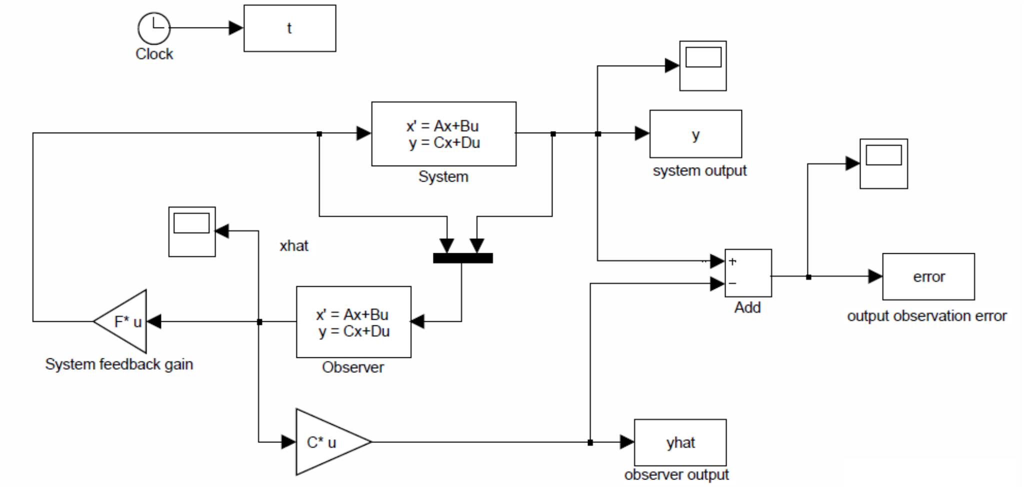

Figure 2: SIMULINK implementation of the system-observer configuration

In Figure 2, 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 maybe observed on the scopes (by clicking on them twice using the left mouse bottom) or plotted in MATLAB using, for example, plot(t,y).

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

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

Using the given dimensions of the state, input, and output variables, respectively given by n, m, and c, we set up the system matrices for the state space SIMULINK blocks (by clicking and opening the blocks and setting the block parameters) 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 in order to be able to run simulation we *must* put something for the initial condition, any vector of dimension *n*, which we denote by x0. The system state space block setup in SIMULINK is:

System State Space Block Setup:

A, B, C, zeros(c,m), x0 (17)

By double clicking on the system state space block in the SIMULINK window using the left mouse, we will be able to setup the system state space data. Namely, a new window will open in which we 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 function, for example for the output observation error, plot(t,error). Such a figure can be further edited using numerous MATLAB figure editing functions.

For the observer, the corresponding matrices should be set using equation (16) as:

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 one is the augmented matrix of two inputs into the observer, see (16), and the third matrix set to identity of dimension n indicates that we want to see (obtain) on the observer output estimates of all n state variables. The forth 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 paper we will discuss a rational choice of the observer initial condition. These matrices can be entered in the SIMULINK state space block by using a left mouse and double clicking on the observer state space block, which will open a new window.

The corresponding MATLAB code is given by

```
>> % System State Space Block Setup
>> n = “dimension of state vector”;
>> % dimension of system;
>> m = “dimension of input vector”;
>> c = “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 = “some 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=”any vector of dimension n”
```

Note that we put for the observer output matrix an identity matrix since we want to see all state variables on the observer output. The observer is an artificial system built by a designer and hence all state variables could be available if the observer is properly designed. We have to specify also the observer initial condition. In general, it can be any vector whose dimension is equal to the dimension of the system state space variables. Engineering experience can help to choose observer's initial condition, but no general guidelines exist. In practice, engineers often set $\hat{x}(0)=0$, [13], so that $e(0)=x(0)-\hat{x}(0)=x(0)$. It was recommended in [13], that partial information about the system initial condition available from the system measurements at the initial time, that is, $y(0)=Cx(0)$, can be used to obtain a value for $\hat{x}(0)$. In such a case, $\hat{x}(0)$ can be obtained via the least square method of [13] as

$$\hat{x}(0)=(C^TC)^{-1}C^Ty(0) \tag{19}$$

where $y(0)$ is a known measured signal at $t=0$.

III. Reduced-Order Observer Design

Consider the linear dynamic system defined in (4) with the corresponding measurements. We have assumed that the output matrix C has full rank equal to c. This means that the output equation $y(t)=Cx(t)$ represents at any time c linearly independent algebraic equations for n unknowns state variables $x(t)$. Note that $y(t)$ is measured system output, and hence a vector of dimension c is known at all times. In this section, we show how to construct an observer of order $r=n-c$ for estimation of the remaining r state space variables (c state variables can be obtained directly from the system measurements $y(t)=Cx(t)$).

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

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

has full rank equal to n. Let us introduce a vector $p(t)$ of dimension r as

$$p(t)=C_1x(t) \tag{21}$$

Putting together (3) and (21), we have

$$\begin{bmatrix} y(t) \\ p(t) \end{bmatrix}=\begin{bmatrix} C \\ C_1 \end{bmatrix}x(t) \quad \Rightarrow$$
$$x(t)=\begin{bmatrix} C \\ C_1 \end{bmatrix}^{-1}\begin{bmatrix} y(t) \\ p(t) \end{bmatrix}=\begin{bmatrix} L & L_1 \end{bmatrix}\begin{bmatrix} y(t) \\ p(t) \end{bmatrix}=Ly(t)+L_1p(t) \tag{22}$$

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 $r=n-c<n$. Before we proceed with construction of a reduced-order observer for $p(t)$, we notice from (22) that the following algebraic relations exist

$$\begin{bmatrix} C \\ C_1 \end{bmatrix}\begin{bmatrix} C \\ C_1 \end{bmatrix}^{-1}=I_n=\begin{bmatrix} C \\ C_1 \end{bmatrix}\begin{bmatrix} L & L_1 \end{bmatrix}=\begin{bmatrix} CL & CL_1 \\ C_1L & C_1L_1 \end{bmatrix}=\begin{bmatrix} I_c & 0 \\ 0 & I_r \end{bmatrix}$$

$$\Rightarrow \quad CL=I_c, \quad C_1L_1=I_r, \quad CL_1=0, \quad C_1L=0 \tag{24}$$

where I_n, I_c, I_r, are unit matrices of corresponding dimensions, with $n=c+r$.

Since from (21) we have $p(t)=C_1x(t)$, the differential equation for $p(t)$ can be easily constructed using the original system differential equation (4) and formula (22), which leads to

$$\dot{p}(t)=\frac{d}{dt}\{p(t)\}=\frac{d}{dt}\{C_1x(t)\}$$
$$=C_1\dot{x}(t)=C_1Ax(t)+C_1Bu(t)=C_1AL_1p(t)+C_1Bu(t) \tag{25}$$

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*) we will find out that $y(t)$ does not carry information about $p(t)$ since in view of (24) we have

$$y(t)=Cx(t)=C(Ly(t)+CL_1p(t))=y(t)+0=y(t) \qquad (26)$$

We get more information about $y(t)$ if we know $\dot{y}(t)$, in which case we have

$$\begin{aligned}\dot{y}(t)&=C\dot{x}(t)=CAx(t)+CBu(t)\\&=CAL_1p(t)+CALy(t)+CBu(t)\end{aligned} \qquad (27)$$

This indicates that $\dot{y}(t)$ carries information about $p(t)$ so that we can construct an observer for $p(t)$ as follows

$$\dot{\hat{p}}(t)=C_1AL_1\hat{p}(t)+C_1ALy(t)+C_1Bu(t)+K_1(\dot{y}(t)-\dot{\hat{y}}(t)) \qquad (28)$$

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

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

Signal differentiation is not a recommended operation since it is very sensitive to noise; it in fact amplifies noise so it should be avoided in all practical applications. We will see from what follows that using an appropriate change of variables we will be able to completely eliminate this request for information about the derivative of system measurements $\dot{y}(t)$.

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

$$\begin{aligned}\dot{\hat{p}}(t)&=C_1AL_1\hat{p}(t)+C_1ALy(t)+C_1Bu(t)\\&+K_1(\dot{y}(t)-CAL_1\hat{p}(t)+CALy(t)+CBu(t))\end{aligned} \qquad (30)$$

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

$$\hat{q}(t)=\hat{p}(t)-K_1y(t) \quad \Rightarrow \quad \dot{\hat{q}}(t)=\dot{\hat{p}}(t)-K_1\dot{y}(t) \qquad (31)$$

we obtain the reduced-order observer for $\hat{q}(t)$ represented by the following linear dynamic system

$$\dot{\hat{q}}(t)=A_q\hat{q}(t)+B_qu(t)+K_qy(t) \qquad (32)$$

where

$$\begin{aligned}A_q&=(C_1-K_1C)AL_1, \quad B_q=(C_1-K_1C)B,\\K_q&=(C_1-K_1C)A(L+L_1K_1)\end{aligned} \qquad (33)$$

Using the observation for $\hat{q}(t)$ obtained from (32)-(33), the reduced order observer estimate is obtained from (31) as $\hat{p}(t)=\hat{q}(t)+K_1y(t)$, which gives the estimate of the original state space variables as

$$\begin{aligned}\hat{x}(t)&=Ly(t)+L_1\hat{p}(t)=Ly(t)+L_1(\hat{q}(t)+K_1y(t))\\&=(L+L_1K_1)y(t)+L_1\hat{q}(t)\end{aligned} \qquad (34)$$

The reduced-order observer is also implemented in SIMULINK. The corresponding matrices defined in (33) and evaluated in MATLAB are entered in the SIMULINK state space block for the reduced-order observer. The obtained system-reduced-observer configuration is presented in Figure 3. Note that we have assumed $\dim\{y(t)\}=l=c$. where $c=\text{rank}\{C\}$. For the reduced-order observer, the corresponding matrices should be set using as:

Observer State Space Block Setup:

Aq, [Bq Kq], eye(n-c), zeros(n-c,m+l); x0hat (35)

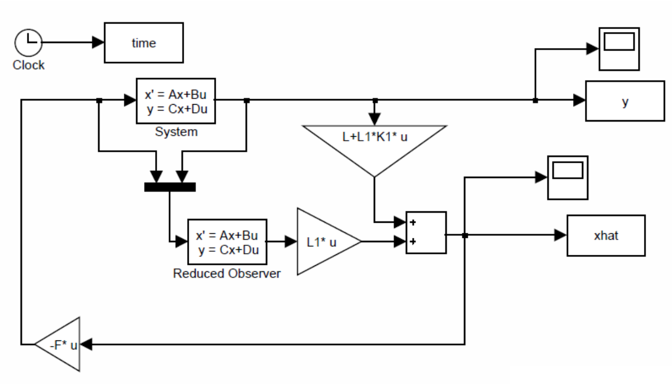

Figure 3: System reduced-order-observer configuration

A. *Setting Reduced-Order-Observer Eigenvalues in the Desired Locations*

We need to appropriately place the reduced-order observer the eigenvalues such that roughly the reduced-order observer be 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

$$A_q=C_1AL_1-K_1CAL_1=(C_1AL_1)-K_1(CAL_1) \qquad (39)$$

To be able to find the reduced-order observer gain K_1 to arbitrarily place the reduced-order observer eigenvalues, the pair $(C_1AL_1, CAL_1))$ must be observable. It can be shown that this condition is satisfied if the original system is observable (the pair (A,C) is observable). We can set the reduced-observer eigenvalues using the following MATLAB code:

```
>> % checking the observability condition
>> O=obsv(C1*A*L1,C*A*L1); rank(O);
>> % rank must be equal to r
>> % finding the closed-loop system poles
>> lambda_sys=eig(A-B*F);
>> % Closed-loop reduced-order observer poles
>> % input desired lambda_obs
>> %(reduced-order observer eigenvalues)
>> K1T=place((C1*A*L1)’, (C*A*L1)’, lambda_obs);
>> K1=K1T’
```

IV. Simulation Example: An Aircraft Full- and Reduced-Order Observer Designs

In this section we consider an aircraft mathematical model taken from [7], whose state space matrices are given by

$$A = \begin{bmatrix} -0.01357 & -32.2 & -46.3 & 0 \\ 0.00012 & 0 & 1.214 & 0 \\ -0.0001212 & 0 & -1.214 & 1 \\ 0.00057 & 0 & -0.1 & -0.6696 \end{bmatrix},$$

$$B = \begin{bmatrix} -0.433 \\ 0.1394 \\ -0.1394 \\ -0.1577 \end{bmatrix}, C = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

The obtained differences between the actual state trajectories and the estimated state trajectories using the full- and reduced-order observers are presented in Figures 4 and 5. In both cases the initial conditions for the observers are obtained using the least square formulas [13].

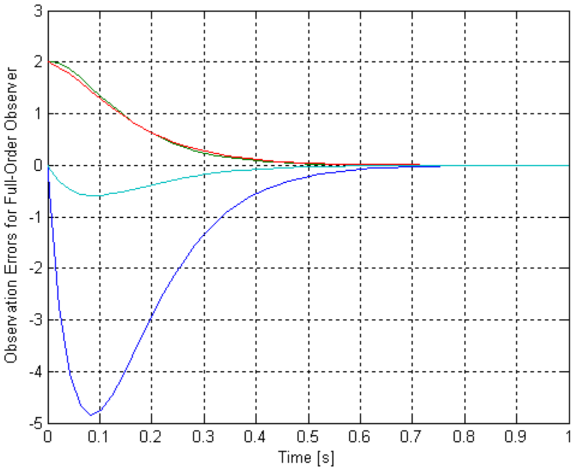

Figure 4: Full-order observer estimation errors

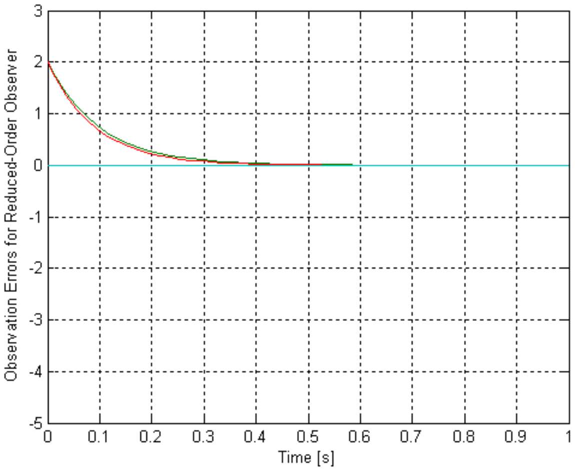

Figure 5: Reduced-order observer estimation errors $x(t)-\hat{x}(t)$ with $\hat{x}(t)$ obtained from (34)

It can be seen from Figures 4 and 5 that the reduced-order observer is superior over the full-order observer. Not just that there are no observation errors for two state variables directly measured, but even those observed by the reduced-order observer are more accurate than the corresponding ones observed by the full-order observer. Note that the eigenvalues for both observers are placed to be of the same speed. The reduced-order observer values are placed at -10, -11 and the full-order observer values are placed at -10, -11, -12, -13. Even more, the reduced-order observer is simpler for implementation since they are dynamical systems of lower order that the original systems. Processing signals that come from lower order dynamic systems is more accurate. Additional advantages of the reduced-order observer are that they require less number of sensors and less number of feedback loops for corresponding feedback control applications.

V. CONCLUSIONS

We have shown in detail how to implement full- and reduced-order observers in MATLAB/Simulink and presented corresponding fundamental derivations and results. We hope that this presentation will motivate 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 more observers in their practice since these days they can be easily implemented using personal computers (PC in the control loop).

REFERENCES

[1] N. Nise, *Control Systems Engineering*, John Wiley and Sons, Hoboken, New Jersey, 2008.

[2] R. Stefani, C. Savant, B. Shahian, and G. Hostetter, *Design of Feedback Control Systems*, Saunders College Publishing, Orlando, Florida, 1994.

[3] G. Franklin, J. Powel, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, Prentice Hall, Upper Saddle River, 2002.

[4] R. Dorf and R. Bishop, *Modern Control Systems*, Pearson Education, Upper Saddle River, New Jersey, 2005.

[5] K. Ogata, *Modern Control Engineering*, Prentice Hall, Upper Saddle River, 2002.

[6] S. Shinners, *Modern Control Systems Theory and Design*, John Wiley and Sons, New York, 1992.

[7] Z. Gajic and M. Lelic, *Modern Control Systems Engineering*, Prentice Hall International, London, 1996.

[8] B. Friedland, *Control Systems Design: An Introduction to State Space Methods*, Dover Publications, 2005.

[9] Z. Gajic, *Linear Dynamic Systems and Signals*, Prentice Hall, 2003.

[10] D. Luenberger, “Observing the state of a linear system,” *IEEE Transactions on Military Electronics*, Vol. 8, 74-80, 1964.

[11] D. Luenberger, “Observers for multivariable systems,” *IEEE Transactions on Automatic Control*, Vol. AC-11, 190-197, 1966.

[12] D. Luenberger, “An introduction to observers,” *IEEE Transactions on Automatic Control*, Vol. AC-16, 596-602, 1971.

[13] C. D. Johnson, “Optimal initial condition for full-order observers,” *International Journal of Control*, Vol. 48, 857-864, 1988.

[14] G. Strang, *Introduction to Linear Algebra*, 4th editon, Wellesley Cambridge Press, 2009.

[15] C-T. Chen, *Linear Systems: Analysis and Design*, 4th edition, Oxford University Press, 2012.