STATE OBSERVER DESIGNS FOR QUARTER-CAR PASSIVE SUSPENSION

Tasya Y. Christnantasari

Department of Engineering Physics, Institut Teknologi Sepuluh Nopember, Surabaya, 60111, Indonesia

ABSTRACT

This paper presents state observer designs for quarter-car passive suspension. Proposed designs correspond to two theories, full-order state observer and observer on closed-loop system. Those observers are used for states and estimation errors observation. Simulation is done using MATLAB and SIMULINK. MATLAB is used to calculate both feedback gain matrix and observer gain matrix whereas SIMULINK is applied to build state space block. Results show that those observers work effectively and fit observer theories. This work may motivate to continue to other steps of observer designs, observer designs for half-car and full car passive suspension.

KEYWORDS

Quarter-Car Passive Suspension, Observer, Closed-Loop System

1. INTRODUCTION

Suspension plays an essential role in isolating vibration due to road surface and improving human comfort and safety. So that, suspension has been widely adopted in many researches since 1980s. Recently, there are three types of suspension that are implemented and tested: passive, semi-active, and active. All of them use either pneumatic or hydraulic operation [1]. Each type has its unique model, characteristic, and components.

Passive suspension is one of types of suspension that widely involved because of its simplicity. Passive suspension consists of damper and spring as its main components. No energy is added to passive suspension system compared to active suspension system. Although it is hard to achieve an optimal condition of ride comfort when using passive suspension, passive suspension is favourable for a lightweight vehicle [2].

There are many analyses concerning passive suspension system model in order to maintain ride comfort.I. Maciejewski et al. (2009) presented modified passive suspension to minimize vibration [3].Hassaan (2004) has examined car dynamics of quarter-car passive suspension to optimize ride comfort [4]. Verros et al. (2005) did design optimization for passive suspension under random road excitation [5].

Many researchers have also tried to achieve optimal condition of passive suspension through several methods. Anirban. C. Mitra et al. (2016)have applied Genetic Algorithm to optimize ride comfort and road holding of passive suspension [6]. Bhargav Gadhvi et al. (2016) have used NSGA-II, SPEA2, and PESA-II for vehicle passive suspension multi-objective optimization [7]. M. Zehsaz et al. (2011) have done optimization based on experimental and numerical methods

DOI:10.5121/ijccms.2020.9301

for tractor's cabin passive suspension [8].Smith and Wang (2004) used inerter in order to optimize passive suspension [9].

Those researches have not covered and responded all challenges of passive suspension system. One of those challenges is state observation process. There are two problems regarding state observation process that should be tackled, state estimation from the measurements such as acceleration or velocity and state observation if actual state is not measurable [10]. Both cases require state observer and/or filter that can generate good estimated state, such as velocity, mass displacement, and suspension deflection. Those estimations can improve ride quality and further can be implemented in other topics.

This paper is focusing on observer designs for passive suspension in quarter-car model based on full-order state observer and closed-loop system theories. Two observer designs will be presented. The first design answers first condition that estimated states construct from available measurement and second design will deal with the second condition where there is no measurable state.

2. Quarter-Car Passive Suspension model

A structure diagram of quarter-car passive suspension model can be seen in Figure 1. The diagram consists of a sprung mass and unsprung mass. Sprung mass represents the mass of quarter of car chassis while unsprung mass reflects the mass of a single wheel assembly. Both masses are connected using a linear spring and viscous damper. Subscript of s refers to sprung elements, subscript u is for unsprung elements, meanwhile subscript r stands for road displacement.

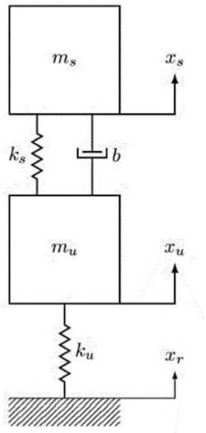

Figure 1. Quarter-car Passive Suspension Model

2.1. Mathematical Model

According to Newton's law, equations of motion of quarter-car passive suspension depicted in Figure 1 are

$$m_s\ddot{x}_s = -b(\dot{x}_s - \dot{x}_u) - k_s(x_s - x_u) \tag{1}$$

$$m_u\ddot{x}_u = b(\dot{x}_s - \dot{x}_u) + k_s(x_s - x_u) - k_u(x_u - x_r) \tag{2}$$

Where variable descriptions are detailed in Table 1.

Table 1. Variables and Descriptions

Variable	Description
m_s	Sprung mass
m_u	Unsprung mass
x_s	Displacement of sprung mass
x_u	Displacement of unsprung mass
x_r	Road displacement
k_s	Spring stiffness
k_u	Tire stiffness
b	Damping coefficient

2.2. State Space Model

This paper uses state space approach to identify input and output of system. State space form is defined as

$$\begin{gathered}\dot{x}(t) = Ax(t) + Bu(t)\\ y(t) = Cx(t) + Du(t)\end{gathered} \tag{3}$$

where are constant matrices. State space form requires variables to define the dynamic system. Those variables are state variable $(x(t))$, output variable$(y(t))$, and input variable$(u(t))$. Quarter-car passive suspension state space form involves four state variables, two output variables, and an input variable. Details of variables is described in Table 2.

Table 2. State Space Variables

Variable	Matrix
State variables	$\begin{bmatrix} x_1\\ x_2\\ x_3\\ x_4 \end{bmatrix} = \begin{bmatrix} x_s\\ \dot{x}_s\\ x_u\\ \dot{x}_u \end{bmatrix}$
Output variables	$\begin{bmatrix} y_1\\ y_2 \end{bmatrix} = \begin{bmatrix} \ddot{x}_s\\ \ddot{x}_u \end{bmatrix}$
Input variable	$[u] = [x_r]$

Therefore, we can rewrite (1) and (2) in state space form as follows

$$\begin{bmatrix} \dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0\\ -\frac{k_s}{m_s} & -\frac{b}{m_s} & \frac{k_s}{m_s} & \frac{b}{m_s}\\ 0 & 0 & 0 & 1\\ \frac{k_s}{m_u} & \frac{b}{m_u} & -\frac{(k_s+k_u)}{m_u} & -\frac{b}{m_u} \end{bmatrix} \begin{bmatrix} x_1\\ x_2\\ x_3\\ x_4 \end{bmatrix} + \begin{bmatrix} 0\\ 0\\ 0\\ \frac{k_u}{m_u} \end{bmatrix} u \tag{4}$$

$$\begin{bmatrix} y_1\\ y_2 \end{bmatrix} = \begin{bmatrix} -\frac{k_s}{m_s} & -\frac{b}{m_s} & \frac{k_s}{m_s} & \frac{b}{m_s}\\ \frac{k_s}{m_u} & \frac{b}{m_u} & -\frac{(k_s+k_u)}{m_u} & -\frac{b}{m_u} \end{bmatrix} \begin{bmatrix} x_1\\ x_2\\ x_3\\ x_4 \end{bmatrix}$$

3. STATE OBSERVER

Observation is method of unmeasurable state variable estimation. It can be tackled with several computer programs (or devices) and one of the is state observer. The theory of observer is originated from Luenberger's works in 1960s to 1970s [11,12,13].According to Luenberger, the system input and output can construct an estimate of system state variables. Therefore, State observer uses data from control variables and measurements of the output to estimate state variables. There are two major kinds of state observer, full-order state observer and reduced-order state observer. Full-order state observer observes all state variables whereas reduced-order state observer estimates fewer than total number of state variables. State observer or simply observer can be designed for either continuous-time system or discrete-time system. This paper uses full-order observer for continuous-time system.

3.1. Full-order State Observer Model

Observer has the same structure with the real system plus feedback term that brings information about observation error [10,14]. Therefore, full-order observer can be obtained from actual systemplus a correction. The correction acts as feedback signal to observer. The correction is based on the system output and estimated output. Actual system is described by

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t)\\ y(t) &= Cx(t) + Du(t)\end{aligned} \tag{5}$$

the observer is designed as

$$\begin{aligned}\dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + L(y(t) - \hat{y}(t))\\ \hat{y}(t) &= C\hat{x}(t) + Du(t)\end{aligned} \tag{6}$$

where L is observer gain. The observer equation (6) is derived from actual system with true state x replaced by estimated state $\hat{x}$. Output equation is also derived from actual system's output with x replaced by $\hat{x}$. As mentioned in aforementioned explanation, observer inputs are u and y and $\hat{x}$ as output. Discrepancy between y and $\hat{y}$ acts as correction and reduces the effect of actual state variables and estimated state variables gap.

To obtain the state space model of observer, we substitute $\hat{y}$ to observer's state equation.

$$\begin{aligned}\dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) + L\big(y(t) - C\hat{x}(t) - Du(t)\big)\\ \dot{\hat{x}}(t) &= (A - LC)\hat{x}(t) + (B - LD)u(t) + Ly(t)\end{aligned} \tag{7}$$

it can be rewritten as

$$\dot{\hat{x}} = [A - LC]\hat{x} + [B - LD \quad L]\begin{bmatrix}u\\ y\end{bmatrix} \tag{8}$$

Although D matrix appears in (8), it has no influence on estimated state produced by observer. It is because correction cancels out D. It is proofed by substituting y and $\hat{y}$ to correction.

$$y(t) - \hat{y}(t) = Cx(t) + Du(t) - C\hat{x}(t) - Du(t) = C(x(t) - \hat{x}(t)) \tag{9}$$

Figure 2 represents real system with full-order state observer with D = 0.

3.2. Estimation Error and Observer Gain

The aim of observer is to produce estimation of true state x. It is reasonable to assume there will be some gap between true state and estimated state but it is hoped that the gap will decrease over time. The gap is called estimation error. Estimation error can be defined as

$$\mathrm{e(t)} = \mathrm{x(t)} - \hat{\mathrm{x}}\mathrm{(t)} \tag{10}$$

then the equation (5) and (6) becomes

$$\begin{gathered} \dot{\mathrm{x}}\mathrm{(t)} - \dot{\hat{\mathrm{x}}}\mathrm{(t)} = \mathrm{Ax(t)} + \mathrm{Bu(t)} - \mathrm{A}\hat{\mathrm{x}}\mathrm{(t)} - \mathrm{Bu(t)} - \mathrm{L(y(t)} - \hat{\mathrm{y}}\mathrm{(t))} \\ \dot{\mathrm{e}}\mathrm{(t)} = \mathrm{(A - LC)e(t)} \end{gathered} \tag{11}$$

Dynamic behaviour of estimation error is determined by eigenvalues of $\mathrm{A - LC}$ matrix. If $\mathrm{A - LC}$ is stable, error will tend to zero and $\hat{x}$ will converge with$x$ for any initial value of x and $\hat{x}$.

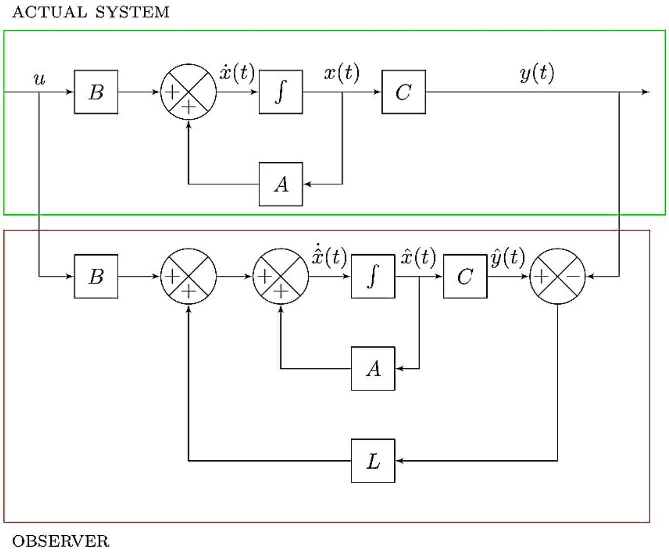

Figure 2. Actual System with State Observer

$\mathrm{A - LC}$ takes crucial part in designing full-state observer. It influences the speed of error convergence. It is dependent to observability of the system. If system (A,C) is completely observable, then the eigenvalues of $\mathrm{A - LC}$ can be chosen arbitrarily [10,15,16]. The chosen eigen values of $\mathrm{A - LC}$ should comply strict rule that complex eigenvalue must be paired with its conjugate. can be determined using various methods, such as Ackermann's formula, matrix transformation, and place in MATLAB [10,17].

3.3. Effect of Addendum of State Observer on Closed-Loop System

If we design a closed-loop system, x(t) is assumed to be available for feedback. In fact, actual state $\mathrm{x(t)}$ may not be measurable. So that, state observer is needed and $\hat{\mathrm{x}}\mathrm{(t)}$ is used for feedback. There will be two-stages process. First stage will determine feedback gain matrix (K) to produce desired characteristic equation and second stage will set observer gain matrix (L) to yield desired observer characteristic equation [10]. Figure 3 represents state observer on closed-loop system.

Input for this system is

$$u(t) = -K\hat{x}(t) \qquad (12)$$

Substituting that input, state equation becomes

$$\dot{x}(t) = Ax(t) - BK\hat{x}(t) = (A - BK)x(t) + BK(x(t) - \hat{x}(t)) \qquad (13)$$

Using definition of estimation error, state equation turns to

$$\dot{x}(t) = ((A - BK)x(t) + BKe(t) \qquad (14)$$

Estimated-state equation becomes

$$\dot{\hat{x}}(t) = (A - LC - BK)\hat{x}(t) + Ly(t) \qquad (15)$$

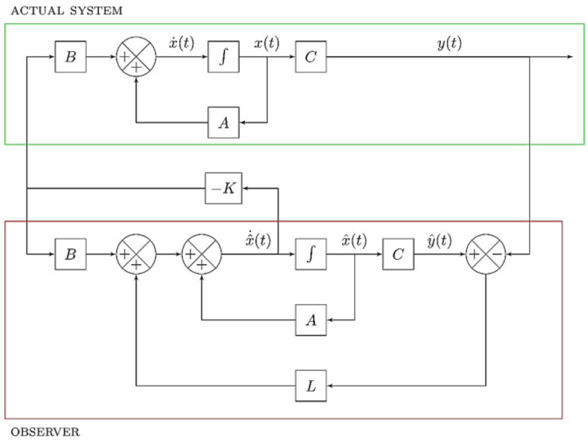

Figure 3. State Observer on Closed-Loop System

That system has two separate eigenvalues, eigenvalues of $A - BK$ and eigenvalues of $A - LC$. Hence, both eigenvalues can be independently and separately placed in the desired locations. If the system (A,B)is controllable, eigenvalues of $A - BK$ can be arbitrarily placed in the plane but it should follow a rule that complex eigenvalues must appear in pairs with its complex conjugate. Feedback gain matrix can be calculated with Ackermann's formula and place in MATLAB.

4. Observer Designs for Quarter-Car Passive Suspension

The parameters used for the observer designs are detailed in Table 3.

Table 3. Simulation Parameters

Parameter	Value	Unit
m_s	100	kg
m_u	10	kg
k_s	5	kN/m
k_u	20	kN/m
b	0.1	kN.s/m

Details of coefficients of state space form are described in Table 4.

Table 4. Details of Coefficients of State Space Form

Coefficient	Value
A	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ -0.05 & -0.001 & 0.05 & 0.001 \\ 0 & 0 & 0 & 1 \\ 0.5 & 0.01 & -2.5 & -0.01 \end{bmatrix}$
B	$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 2 \end{bmatrix}$
C	$\begin{bmatrix} -0.05 & -0.001 & 0.05 & 0.001 \\ 0.5 & 0.01 & -2.5 & -0.01 \end{bmatrix}$
D	0

This paper represents two designs, full state observer and observer on closed-loop system. Step input as representative of road displacement will be applied in first design. Both designs require observer gain matrix (L) which is computed in MATLAB. Another design also needs feedback gain matrix (K) which is calculated in MATLAB. These systems are simulated in SIMULINK. Initial values of actual system and observer and other simulation details are explained in Table 5.

Table 5. Simulation Details

Simulation Detail	Value
Initial values of actual system	$\begin{bmatrix} 0.8 \\ 0.8 \\ 0.8 \\ 0.8 \end{bmatrix}$
Initial values of observer	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$
Eigenvalues of $A - LC$	$\lambda_1 = -5 - j1$ $\lambda_2 = -5 + j1$ $\lambda_3 = -4 - j1$ $\lambda_4 = -4 + j1$
Eigenvalues of $A - BK$	$\lambda_1 = -20 + j2$ $\lambda_2 = -20 - j2$ $\lambda_3 = -30 - j3$ $\lambda_4 = -30 + j3$

These observer designs use block diagrams depicted in Figure 3 and 4. Figure 5 and 6 show SIMULINK configurations for full-order state observer design and full state observer on closed-loop system respectively.

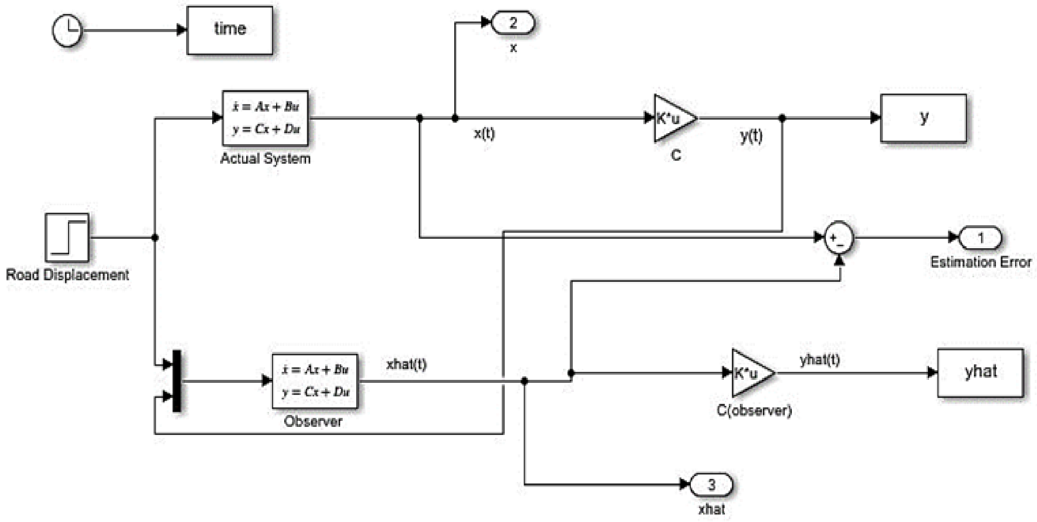

Figure 5. Simulink Configuration for Full-Order State Observer

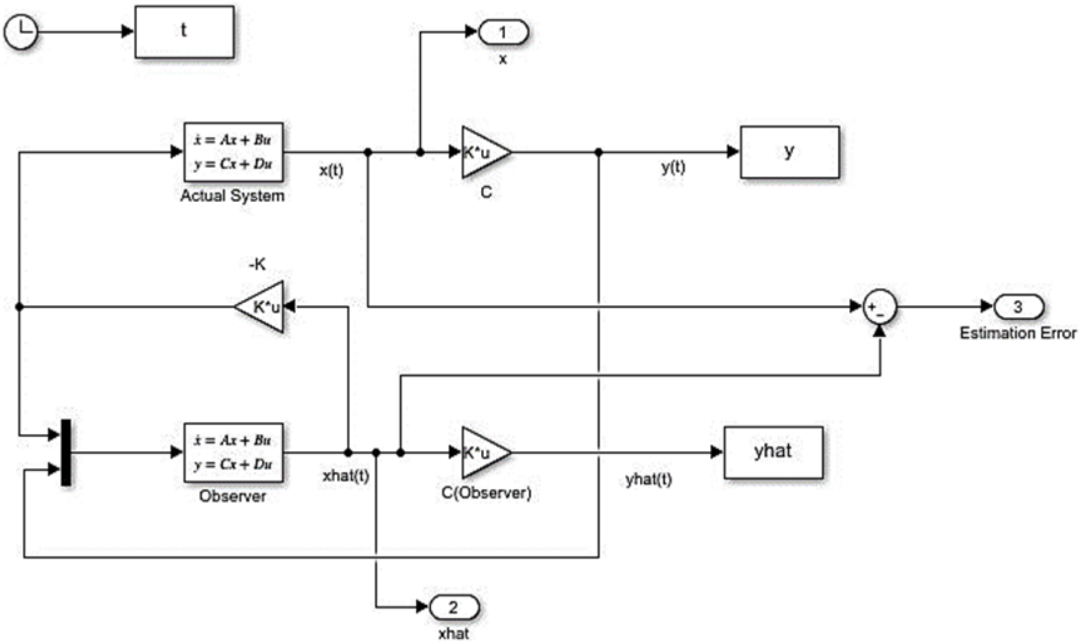

Figure 6. Simulink Configuration for Full State Observer on Closed-Loop System

5. SIMULATION RESULT

Simulink simulation yielded the following results. Each state produces different result although all of them fit observer theory. Figure 4 and 6 present comparison of actual state and estimated state of full-order state observer and observer on closed-loop system respectively. Figure 5 and 7 show the estimation errors of full-order state observer and observer on closed loop system.

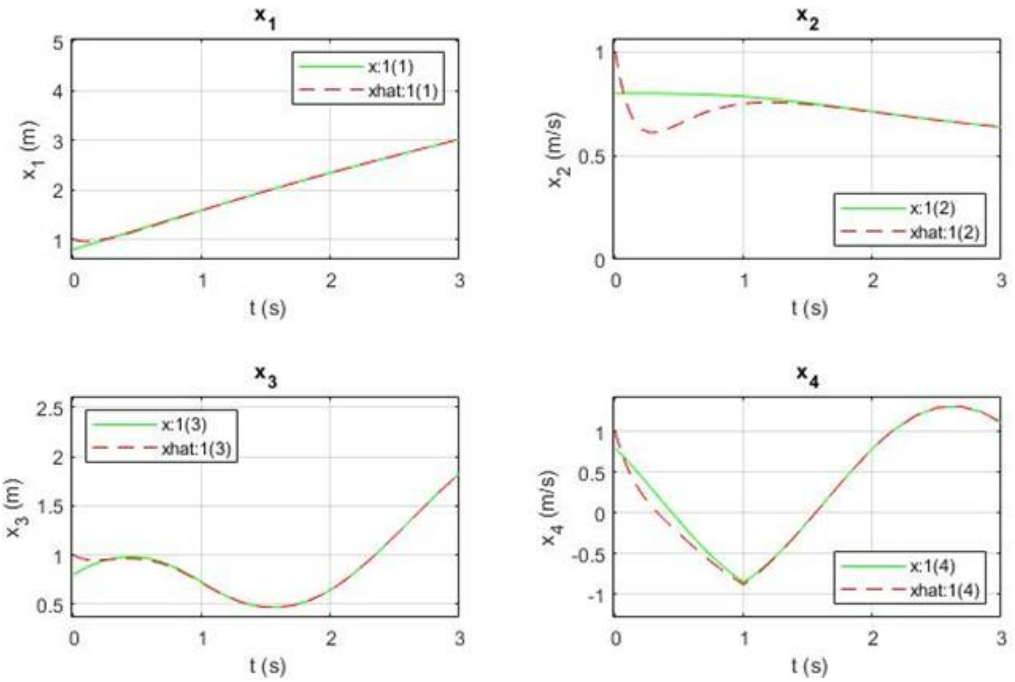

Figure 7. Actual and Estimated States of Full-Order State Observer

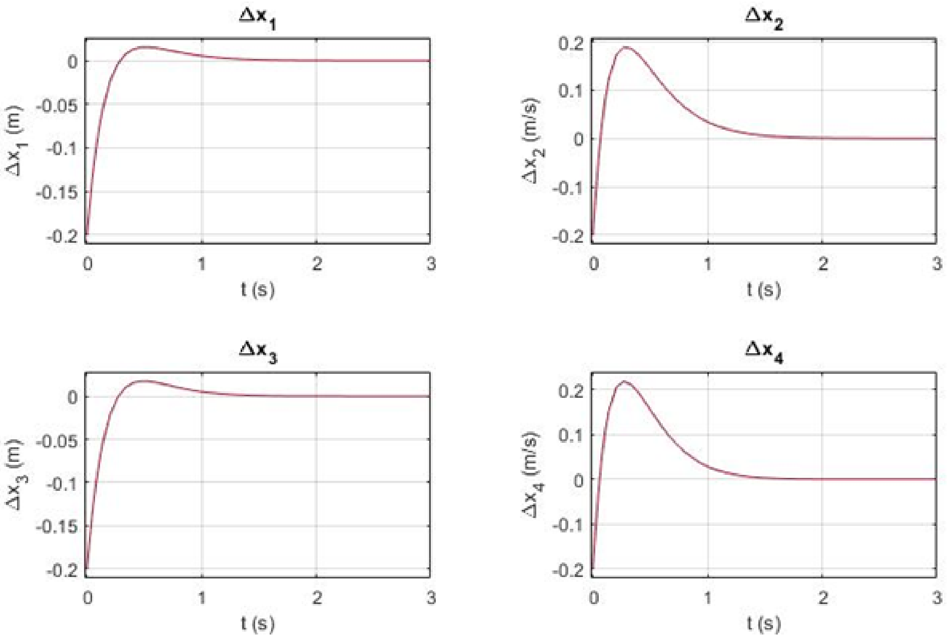

Figure 8. Estimation Errors of Full-Order State Observer

Results show that estimated states converge to actual states with certain speed and estimation errors turn to zero over time. First estimated state ($\widehat{x_1}$) takes 1.2s. Second estimated state ($\widehat{x_2}$) coincides with actual state at 1.6s. Third estimated state ($\widehat{x_3}$) needs 1.2s to coincide with its real state. Fourth estimated state ($\widehat{x_4}$) converges to actual state at 1.4s. Estimation errors need identical time with state convergence to converge to zero. This speed depends on eigenvalues of $A - LC$.

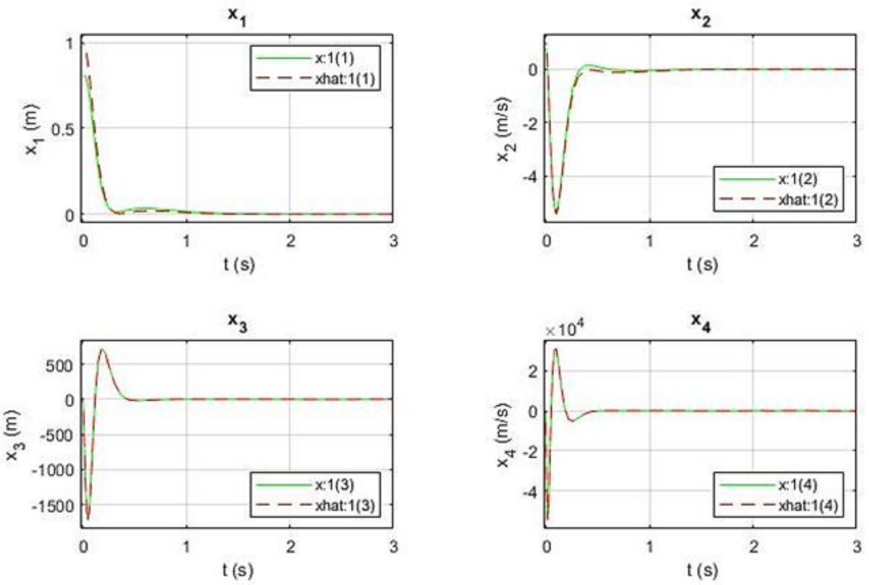

Figure 9. Actual and Estimated State of Full State Observer on Closed-Loop System

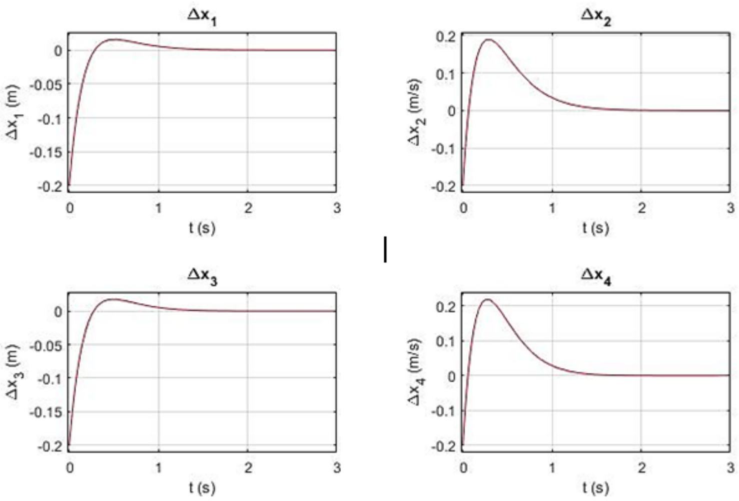

Figure 10. Estimation Errors of Full State Observer on Closed-Loop System

Results describe that each estimated to state converges its actual state yet all of them takes different time. $\widehat{x_1}$converges $x_1$at 1.2s.$\widehat{x_2}$ takes 1.6s. $\widehat{x_3}$ coincides with actual state x_3 at 1.2s and $\widehat{x_4}$ requires 1.4s to converge to its actual state. They mean each state has disparate speed convergence. Estimation errors also tend to zero over time as previous result. Both estimation errors produce identical graphics.

6. CONCLUSION

In this paper, observer for quarter-car passive suspension is designed based on full-order state observer and observer on closed-loop system. Quarter-car passive suspension model is built and transformed into state space model and then full-order observer and closed-loop with observer are applied to estimate each state and observe estimation error. Results show that each estimated state coincides with actual state and estimation errors converge to zero. These results fit observer theories. The obtained results may inspire and become a basic for the next step, observer design for half-car and full-car passive suspension.

ACKNOWLEDGEMENTS

I am thankful to all of my lecturers for providing guidance regarding this topic.

REFERENCES

[1] I. Martins, J. Esteves, F. P. da Silva, P. Verdelho,(2015) "Electromagnetics hybrid active passive vehicle suspension system", Technical University of Lisbon, Lisbon, Portugal.

[2] Jamali, M.S, Ismail, K.A., Taha, Z., Aiman, M.F., (2017) "Development of MATLAB. Simulink Model for DynamicsAnalysis of Passive Suspension System for Lightweight Vehicle", Journal of Physics: Conference Series, Vol. 908, p. 012066.

[3] I. Maciejewski, L. Meyer, T. Krzyzynski, (2009) "Modelling and multi-criteria optimisation of passive seat suspension vibro-isolating properties", Journal of Sound and Vibration, Volume 324, Issues 3-5, pp 520-538.

[4] Galal A. H., (2014) "Car Dynamics using Quarter Model and Passive Suspension,Part I: Effect of Suspension Damping and Car Speed", International Journal of Computer Techniques, Volume 1, Issue 2.

[5] G. Verros, S. Natsiavas, C. Papadimitriou, (2005) "Design Optimization of Quarter-car Models with Passive and Semi-active Suspensions under Random Road Excitation", Journal of Vibration and Control, Vol. 11, pp 581–606.

[6] Anirban. C. Mitra, Gourav. J. Desai, Saaish. R. Patwardhan, Parag H. Shirke, Waseem M. H. Kurne, Nilotpal Banerjee, (2016) "Optimization of Passive Vehicle Suspension System by Genetic Algorithm", Procedia Engineering, Vol. 144, pp 1158 – 1166.

[7] B. Gadhvi, V. Savsani, V. Patel, (2016) "Multi-Objective Optimization of Vehicle Passive Suspension System using NSGA-II, SPEA2 and PESA-II", Procedia Technology, Vol. 23, pp 361–368.

[8] M. Zehsaz, M.H. Sadeghi, M.M. Ettefagh, F. Shams, (2011) "Tractor Cabin's Passive Suspension Parameters Optimization Via Experimental and Numerical Methods", Journal of Terramechanics, Vol. 48, pp 439–450.

[9] Malcolm C. Smith, Fu-Cheng Wang, (2004) "Performance Benefits in Passive Vehicle Suspensions Employing Inerters, Vehicle System Dynamics", International Journal of Vehicle Mechanics and Mobility, Volume 42, Issue 4, pp 235-257.

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

[11] D. Luenberger, (1964) "Observing the state of a linear system", IEEE *Transactions on Military Electronics*, Vol. 8, pp 74-80.

[12] D. Luenberger, (1966) "Observers for multivariable systems", IEEE *Transactions on Automatic Control*, Vol. AC-11, pp 190-197.

[13] D. Luenberger, (1971) "An introduction to observers", IEEE *Transactions on Automatic Control*, Vol. AC-16, pp 596-602.

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

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

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

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

AUTHOR

Tasya Christnantasari is an undergraduate engineering physics student at Institut Teknologi Sepuluh Nopember. She specializes in control theory and is working on research regarding fault-tolerant control and filtering methods.

APPENDIX: MATLAB Code

```matlab
%Observer designs for quarter-car passive suspension
%Defining parameters
ms = 100; %sprung mass
mu = 10; %unsprung mass
ks = 5; %spring stiffness
b = 0.1; %damping constant
ku = 20; %tire stiffness
%Defining state-space model
A = [0 1 0 0;-ks/ms -b/ms ks/ms b/ms;0 0 0 1;ks/mu b/mu -(ks+ku)/mu
-b/mu];
B = [0;0;0;ku/mu];
C = [-ks/ms -b/ms ks/ms b/ms;ks/mu b/mu -(ks+ku)/mu -b/mu];
D = 0;
x0 = [0.8;0.8;0.8;0.8]; %initial state for actual system
x0hat = [1;1;1;1]; %initial state for observer
%Checking the observability
R = [C' A'*C' (A'^2)*C' (A'^3)*C'];
if rank(R) == rank (A)
    disp('system is observable')
else
   disp('system is not observable')
end
%Checking the controlability
M = [B A*B A^2*B A^3*B];
if rank(M) == rank (A)
    disp('system is controlable ')
else
    disp('system is not controlable')
end
poles = [-20+2i -20-2i -30+3i -30-3i]; %A-BK eigenvalues
K = place(A,B,poles); %Feedback gain matrix
poles1 = [-5-1i -5+1i -4+1i -4-1i]; %A-LC eigenvalues
Lt = place(A',C',poles1);
L = Lt'; %Observer gain matrix
```