Paper from Proceeding of the National Conference "Science in Media 2012" Organized by YMCA University of Science and Technology, Faridabad, Haryana (India) December 3rd -4th 2012 Published by
IJMRS's International Journal of Engineering Sciences, ISSN (Online): 2277-9698
www.ijmrs.com

Performance improvement of bus suspension using PID controller

Swati Gaur and Sheilza Aggarwal, YMCA University of Science and Technology, Faridabad, India

***Abstract*—This paper deals with the nonlinear system of Bus Suspension system. This paper presents identification and modelling of bus suspension system with the disturbance. When the suspension system is designed, a 1/4 bus model (one of the four wheels) is used to simplify the problem to a one dimensional spring-damper system. The open loop behaviour of the system on the basis of time and frequency response is analyzed. State space modelling of the device is done to analyze its state space behaviour. A good bus suspension system should have satisfactory road holding ability, while still providing comfort when riding over bumps and holes in the road. When bus experiences any road disturbance such that pot holes, cracks, and uneven pavement, the bus body should not have large oscillations or oscillations should dissipate quickly. The system should have short settling time and also have the ability to absorb all the bumping. To achieve all these objectives closed loop system is required. To design a controller, bus suspension system is linearized. Despite continuous advancement in control theory, Proportional Integral Derivative (PID) controller is the most popular technique to control any process. In this paper, Proportional- Integral – Derivative (PID) Controller is also designed and tuned to give the smooth response for the bus suspension system. System performances for the desired parameters in closed loop are investigated. The simulation and implementation of the controllers are done using MATLAB/SIMULINK software.**

***Keywords*— Bus suspension system, dynamic modelling, PID control, state-space model, MATLAB /Simulink.**

I. Introduction

Today, a struggling race is taking place among the automotive industry to produce highly developed suspension models. One of the performance requirements is advanced suspension systems which prevent the road disturbances to affect the passenger comfort while increasing riding capabilities and performing a smooth drive. The main purpose of this system is to increase the comfort of vehicle occupants (passengers and drivers), to maintain the contact between the tire and the road surface and to eliminate (minimize) dynamic forces which act on the load bearing vehicle structure and road surface along which the vehicle is moving. While the purpose of the suspension system is to provide a smooth ride in the bus and to help maintain control of the vehicle over rough terrain or in case of sudden stops, increasing ride comfort results in larger suspension stroke and smaller damping in the wheel-hop mode [1]. Numerous applications of different control strategies have been proposed to overcome these suspension problems. Many active control strategies such as Linear Quadratic Gaussian (LQG) control, adaptive control, and nonlinear control are developed and proposed so as to manage the occurring problems [2-4]. Among the recent control methods, PID control methods grab nowadays the attention of many researchers. A PID has excellent capability in a nonlinear system description and is particularly suitable for the complex and uncertain systems.

II. System Identification and Modelling

A mathematical model is an abstract model that uses mathematical language to describe the behaviours of a system. From the bus suspension system model, we can directly get the dynamic equation by using the Newton's law. Then, this dynamic equation will be transfer into the Matlab to get the transfer function using the built in function. In this project, there are two outputs because of the mass of the system plus the bus mass.

State-Space Model

The one dimensional spring-mass-damper system given in Figure 1

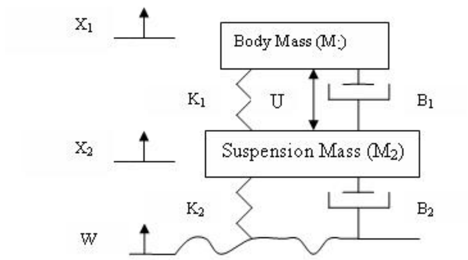

Fig. 1 ¼ Model of Bus Suspension system

The followings are constants and variables of the system we are going to design:

- M_1= 2500 kg, (body mass)
- M_2 = 320 kg, (suspension mass)

- K_1 =80,000N/m, (spring constant of suspension system)
- K_2 = 500,000 N/m, (spring constant of wheel and tire)
- B_1 =350Ns/m, (damping constant of suspension system)
- B_2=15,020 Ns/m, (damping constant of wheel and tire)
- U = force from the controller

To derive the dynamic equations of this system, we used Newton's second law of motion and the equations below are presented.

$$M_1\ddot{X}_1 + B_1(\dot{X}_1 - \dot{X}_2) + K_1(X_1 - X_2) = U \qquad (1)$$

$$M_2\ddot{X}_2 = B_1(\dot{X}_1 - \dot{X}_2) + K_1(X_1 - X_2) + B_2(\dot{W} - \dot{X}_2) + K_2(W - X_2) - U \qquad (2)$$

To transform the motion equations of the quarter-bus model into a state-space model, the equation (4), including variable vector, input vector and the disturbance vector is formed after some algebraic operations.

$$\dot{X} = [A][X] + [B]W, Y = [0010]\begin{bmatrix} X_1 \\ \dot{X}_1 \\ Y_1 \\ Y_2 \end{bmatrix} + [00]\begin{bmatrix} U \\ W \end{bmatrix}, Y = [C][X] + [D]W \qquad (3)$$

$$\begin{bmatrix} \dot{X}_1 \\ \ddot{X}_1 \\ \dot{Y}_1 \\ \dot{Y}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-B_1B_2}{M_1M_2} & 0 & \left(\frac{B_1}{M_1}\left(\frac{B_1}{M_1} + \frac{B_1}{M_2} + \frac{B_2}{M_2}\right)\right) - \frac{K_1}{M_1}\right) & \frac{-B_1}{M_1} \\ \frac{B_2}{M_2} & 0 & -\left(\frac{B_1}{M_1} + \frac{B_1}{M_2} + \frac{B_2}{M_2}\right) & 1 \\ \frac{K_2}{M_1} & 0 & -\left(\frac{K_1}{M_1} + \frac{K_1}{M_2} + \frac{K_2}{M_2}\right) & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ \dot{X}_1 \\ Y_1 \\ Y_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \frac{1}{M_1} & \frac{B_1B_2}{M_1M_2} \\ 0 & \frac{-B_2}{M_2} \\ \left(\frac{1}{M_1} + \frac{1}{M_2}\right) & \frac{-K_2}{M_2} \end{bmatrix} \begin{bmatrix} U \\ W \end{bmatrix} \qquad (4)$$

Since the distance X_1-W is very difficult to measure, and the deformation of the tire (X_2- W) is negligible, we will use the distance X_1-X_2 instead of X_1-W as the output in our problem. The road disturbance (W) in this problem will be simulated by a step input. This step could represent the bus coming out of a pothole.

III. OPEN LOOP ANALYSIS

We have used MATLAB to display how the original open-loop system performs without any feedback control. We see the response of unit step actuated force input and unit step disturbance input.

From the graph of the open-loop response for a unit step actuated force, we can see that the system is under-damped. People sitting in the bus will feel very small amount of oscillation and the steady-state error is about 0.013 mm. Moreover, the bus takes very unacceptably long time for it to reach the steady state. The solution to this problem is to add a controller into the system's block diagram to improve the performance.

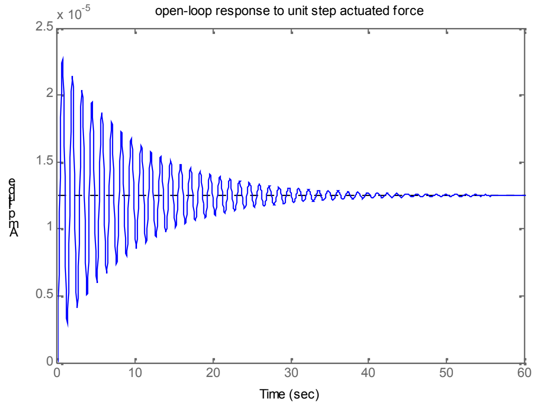

Fig. 2 open-loop response to unit step actuated force

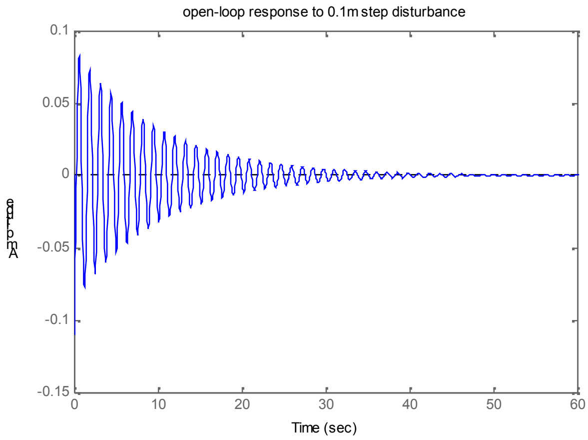

Fig. 3 open-loop response to 0.1m step disturbance

IV. CLOSED LOOP RESPONSE

The schematic of the closed-loop system is the following:

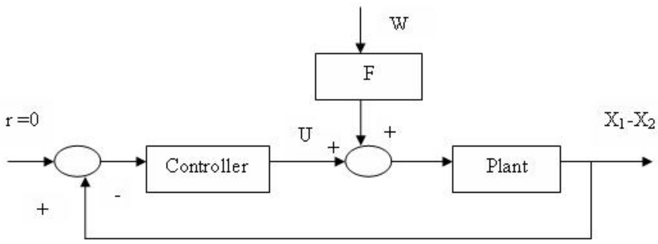

Fig. 4 Closed-loop system with disturbance.

A. *Controller: Proportional-Integral-Derivative Controller (PID controller)*

A proportional-integral-derivative controller (PID controller) is a generic control loop feedback mechanism widely used in industrial control systems - a PID is the most commonly used feedback controller. A PID controller calculates an "error" value as the difference between a measured process variable and a desired set point. The controller attempts to minimize the error by adjusting the process control inputs. In the absence of knowledge of the underlying process, PID controllers are the best controllers.

The PID controller is probably the most-used feedback control design. PID is an acronym for Proportional-Integral-Derivative, referring to the three terms operating on the error signal to produce a control signal. If U(t) is the control signal sent to the system, y(t) is the measured output and r(t) is the desired output, and tracking error e(t) =r(t)-y(t), a PID Controller has the general form

$$u(t) = K_P e(t) + K_I e(t)dt + K_D \frac{d}{dt} e(t) \qquad 5$$

The desired closed loop dynamics is obtained by adjusting the three parameters KP, KI and KD, often iteratively by "tuning" and without specific knowledge of a plant model.

The PID controller calculation involves three separate parameters, and is accordingly sometimes called three-term control: the proportional, the integral and derivative values, denoted P, I, and D. The proportional value determines the reaction to the current error, the integral value determines the reaction based on the sum of recent errors, and the derivative value determines the reaction based on the rate at which the error has been changing. The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve or the power supply of a heating element.

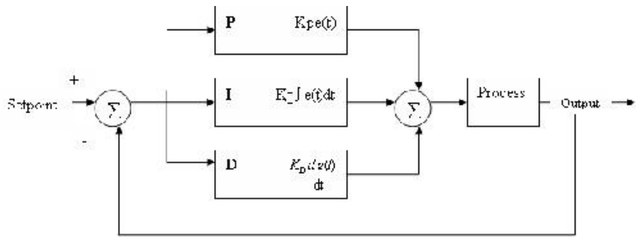

Fig 5 Block Diagram of PID Controller

B. *Tuning*

Tuning a control loop is the adjustment of its control parameters i.e. gain/proportional band, integral gain/reset, derivative gain/rate to the optimum values for the desired control response. Stability is a basic requirement, but beyond that, different systems have different behavior, different applications have different requirements, and some desiderata conflict. Further, some processes have a degree of non-linearity and so parameters that work well at full-load conditions don't work when the process is starting up from no load; this can be corrected by gain scheduling. PID controllers often provide acceptable control even in the absence of tuning, but performance can generally be improved by careful tuning, and performance may be unacceptable with poor tuning. PID tuning is a difficult problem, even though there are only three parameters and in principle is simple to describe, because it must satisfy complex criteria within the limitations of PID control. There are accordingly various methods for loop tuning, and more sophisticated techniques are the subject of patents; this section describes some traditional manual methods for loop tuning.

C. *Tuning Methods*

There are several methods for tuning a PID loop. The most effective methods generally involve the development of some form of process model, then choosing P, I, and D based on the dynamic model parameters. Manual tuning methods can be relatively inefficient, particularly if the loops have response times on the order of minutes or longer. The choice of method will depend largely on whether or not the loop can be taken "offline" for tuning, and the response time of the system. If the system can be taken offline, the best tuning method often involves subjecting the system to a step change in input, measuring the output as a function of time, and using this response to determine the control parameters.

Manual Tuning

If the system must remain online, one tuning method is to first set KI and KD values to zero. Increase the Kp until the output of the loop oscillates, then the KP should be set to approximately half of that value for a "quarter amplitude decay" type response. Then increase KI until any offset is correct in sufficient time for the process. However, too much KI will cause instability. Finally, increase KD, if required, until the loop is acceptably quick to reach its reference after a load disturbance. However, too much KD will cause excessive response and overshoot. A fast PID loop tuning usually overshoots slightly to reach the set point more quickly; however, some systems cannot accept overshoot, in which case an over-damped closed-loop system is required, which will require a KP setting significantly less than half that of the KP setting causing oscillation.

Ziegler-Nichols Method

Another heuristic tuning method is formally known as the Ziegler-Nichols method, introduced by John G. Ziegler and Nathaniel B. Nichols. As in the method above, the KI and KD gains are first set to zero. The P gain is increased until it reaches the ultimate gain, Ku, at which the output of the loop starts to oscillate. Ku and the oscillation period Pu are used to set the gains as shown:

TABLE I
EFFECTS OF INCREASING A PARAMETER INDEPENDENTLY

Control Type	K_p	K_I	K_D
P	0.50 K_u	--------	-------
PI	0.45 K_u	1.2 K_p/P_u	--------
PID	0.60 K_u	2 K_p/P_u	$K_p P_u/8$

Where,
K_u = Ultimate Gain = 1/M
M= amplitude ratio of system's response at crossover frequency
P_u= Ultimate period = $2\pi / \omega co$
ωco= system's crossover frequency.

The closed - loop Ziegler-Nichols method consist of following steps:

1. With P-only closed loop control, increase the magnitude of the proportional gain until the closed loop is in a continuous oscillation. For slightly larger values of controller gain, the closed loop system is unstable, while the slightly lower values the system is stable.

2. The value of controller proportional gain that causes the continuous oscillation is called the critical gain, K_u. The peak-to - peak period is called critical period P_u.

3. Depending upon controller chosen, P, PI, or PID, use the value in table1 for tuning parameters, based on the critical gain and period.

Response

we should see the response (X1-X2) to a step W

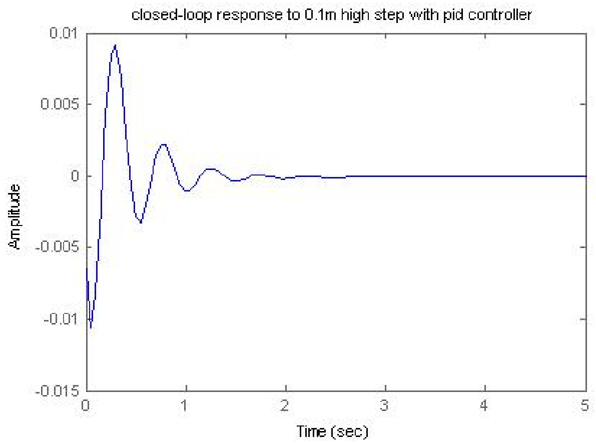

Fig 6 closed-loop response to step with PID controller

we can see that the system has larger damping than required, but the settling time is very short. This response still doesn't satisfy the overshoot requirement.

This can be rectified by manual tuning to find better response, figure 7 where the maximum overshoot is approximately 0.0048 m and the settling time is about 1.5 seconds.

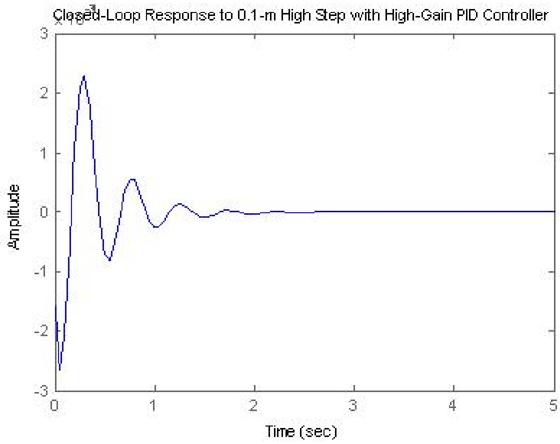

Fig 7 closed-loop response to step with high gain

V. CONCLUSIONS

In this study, a PID controller is designed and employed for controlling an active suspension system of a ¼ bus model. The proposed model is aimed to developed and carry the response of PID controller up to a better level by simply changing only the gains of a PID controller using Manual Tuning method.

REFERENCES

[1] Mathworks Matlab. [Online] Available: http://www.mathworks.com/products/matlab/
[2] Mathworks. Simulink. [Online]. Available: www.mathworks.com/products/simulink/
[3] A. Tewari, Modern Control Design with Matlab and Simulink. John Wiley & Sons, Ltd, 2002.
[4] N. S. Nise,Control Systems Engineering, 6th ed. John Wiley & Sons, Inc, 2011.
[5] K.Ogata, Modern Control Engineering, 5th ed. Pearson Inc, 2010.
[6] U. Itkis, Control Systems of Variable Structure Wiley, 1976.
[7] H. Chen, Z. -Y. Liu, P.-Y. Sun , "Application of Constrained H_Control to Active Suspension Systems on Half-Car Models", Journal of Dynamic Systems, Measurement, and Control, Vol. 127 / 353, Sep 2005.
[8] Gordon, T. J., Marsh, C., and Milsted, M. G., "A Comparison of Adaptive LQG and Nonlinear Controllers for Vehicle Suspension Systems," Veh. Syst. Dyn., 20, 1991,pp. 321-340.

[9] Alleyne, A., and Hedrick, J. K.,"Nonlinear Adaptive Control of Active Suspensions," IEEE Trans. Control System. Technology., 3(1), 1995, pp.94-101.
[10] Ben Gaid, M., Cela, A.,Kocik, R., "Distributed control of a car suspension system," COSI - ESIEE - Cit´e Descartes,
[11] Bennett, Stuart (1993). A history of control engineering, 1930-1955. IET. p. p. 48. ISBN 978-0-86341-299-8.
[12] Bennett, Stuart (November 1984). "Nicholas Minorsky and the automatic steering of ships". IEEE Control Systems Magazine 4 (4): 10–15. doi: 10.1109/MCS.1984. 1104827. ISSN 0272-1708.
[13] "A Brief Building Automation History". Retrieved 2011-04-04.
[14] Bennett, Stuart (June 1986). A history of control engineering, 1800-1930. IET. pp. 142–148. ISBN 978-0-86341-047-5.
[15] Jinghua Zhong (Spring 2006). PID Controller Tuning: A Short Tutorial. Retrieved 2011-04-04.
[16] Ang, K.H., Chong, G.C.Y., and Li, Y. (2005). PID control system analysis, design, and technology, IEEE Trans Control Systems Tech, 13(4), pp.559-576. http://eprints.gla.ac.uk/3817/
[17] Li, Y. and Ang, K.H. and Chong, G.C.Y. (2006) PID control system analysis and design - Problems, remedies, and future directions. IEEE Control Systems Magazine, 26 (1). pp. 32-41. ISSN 0272-1708.
[18] Yang, T. (June 2005). "Architectures of Computational Verb Controllers: Towards a New Paradigm of Intelligent Control". International Journal of Computational Cognition (Yang's Scientific Press) 3 (2): 74–101.