Visualization of Active Suspension by Robust Controller in Virtual Reality Toolbox

Štěpán Ožana
Faculty of Electrical Engineering and Computer Science
Department of Measurement and Control
17.listopadu 15, Ostrava, Czech Republic
stepan.ozana@vsb.cz

Abstract

This paper deals with design and visualization of H-∞ robust controller. It gives a brief mathematical description of physical model of suspension system of a car which was chosen as a typical example of robust control use. Therefore, first a quarter car model is introduced. Then comes the main part of solution of this problem, that is detailed design of the controller. Finally, based on real values of parameters together with a vivid visualization using a powerful environment of Virtual Reality Toolbox, it represents a useful aid in education of Modern Control Theory at the Department of Measurement and Control.

1 Introduction

Development of control methods for passive and active suspension systems have been studied for last 30 years. This paper describes one of methods of robust control design, H-∞ control, demonstrated on a suspension system of a car which is a typical system with so called structured uncertainty. That's why this method is preferred to other ones such as LQR, LQG, predictive or optimal control.

Well designed suspension system has to combine and meet two contradictory requirements, which is ride quality (handling) and passenger's comfort. This can be done by a passive suspension systems. On the other hand, active suspension can meet all the three performance aspects (comfort, handling and rattle space).

2 Quarter Car Model

The mathematic model which represents the dynamics of the system can be obtained from the scheme on Fig.1.

$$m_b z_b^{''} = f_a - k_1 \left(z_b - z_w\right) - c_s \left(z_b^{'} - z_w^{'}\right)$$
$$m_w z_w^{''} = -f_a + k_1 \left(z_b - z_w\right) - k_2 \left(z_w - z_r\right) + c_s \left(z_b^{'} - z_w^{'}\right)$$

m_b...sprung mass
m_w...unsprung mass
c_s...suspension damper
k_1...suspension spring constant
k_2...tire spring constant
z_b...chassis level (height)
z_w...wheel assembly level
z_r...road profile

The sprung mass m_b represents the car chassis, while the unsprung mass m_w represents the wheel assembly. The spring, k_1, and damper, c_s, represent a passive spring and shock absorber that are placed between the car body and the wheel assembly, while the spring k_2 serves to model the compressibility of the pneumatic tire. The variables z_b, z_w, and z_r are the car body travel, the wheel travel, and the road disturbance, respectively. The force f_a applied between the sprung and unsprung masses, is controlled by feedback and represents the active component of the suspension system.

2.1 State-space representation

State-space description of the model is given by matrices A,B,C,D in the form of

$$x^{'} = Ax + Bu$$
$$y = Cx + Du$$

The system has two inputs: road profile z_r and actuating force f_a.

$$u = \begin{bmatrix} z_r \\ f_a \end{bmatrix}$$

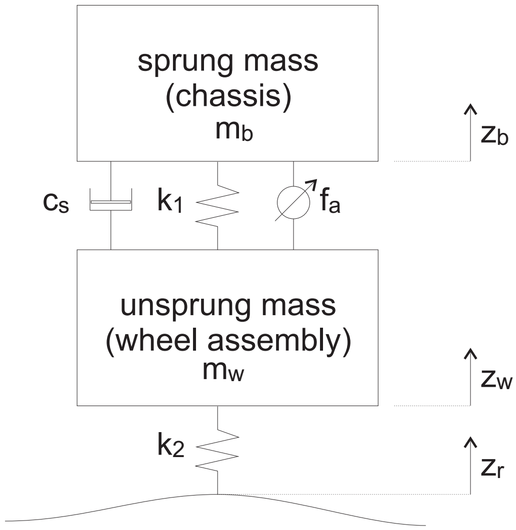

Figure 1. Quarter car model

The state variables of the systems are chosen now:

$$x_1 = z_b, x_2 = z_w, x_3 = z_b^{'}, x_4 = z_w^{'}$$

The equations can be rewritten in the form of

$$m_b x_3^{'} = f_a - k_1 (x_1 - x_2) - c_s (x_3 - x_4)$$
$$m_w x_4^{'} = -f_a + k_1 (x_1 - x_2) - k_2 (x_2 - z_r) + c_s (x_3 - x_4)$$

Then output quantities must be determined. In real operation, theese quantities could be measured:
$y_1 = x_1 - z_r$...distance between chassis and the road
$y_2 = x_1 - x_2$...distance between chassis and wheel center
$y_3 = x_3^{'}$...vertical acceleration of chassis
The matrices A,B,C,D are then:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-k_1}{m_b} & \frac{k_1}{m_b} & \frac{-c_s}{m_b} & \frac{c_s}{m_b} \\ \frac{k_1}{m_w} & \frac{-k_1-k_2}{m_w} & \frac{c_s}{m_w} & \frac{-c_s}{m_w} \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{m_b} \\ \frac{k_2}{m_w} & \frac{-1}{m_w} \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ \frac{-k_1}{m_b} & \frac{k_1}{m_b} & \frac{-c_s}{m_b} & \frac{c_s}{m_b} \end{bmatrix}, D = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & \frac{1}{m_b} \end{bmatrix}$$

3 Design of a controller

The purpose of active suspension system of a car is to provide better features than passive suspension, mainly better traveller's comfort and stability of a car on the road. These requirements speak in contradictions and a compromise must be found during the design of a controller.

P stands for the plant, K for controller, w for external quantities(control quantity, external influences – noise, measurement errors,disturbances), u for actuating quantity,

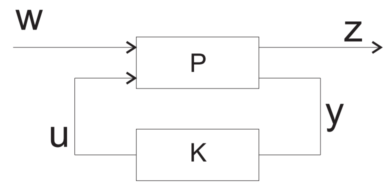

Figure 2. Basic block scheme for H-∞ design

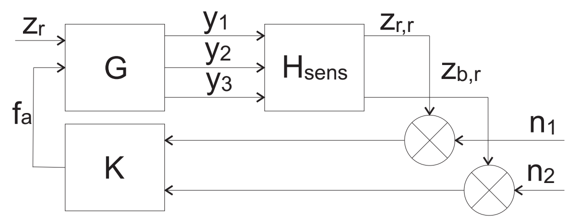

Figure 3. Block scheme for realization of a controller

z for error outputs (to be minimized) and y for measured quantities, see Fig.2. Using H-∞ it's possible to find a stabilizing robust controller, which is optimal and minimizes error outputs z. Extended form must be used to describe the plant P: $x^{'} = Ax + B_1 w + B_2 u$

$$z = C_1 x + D_{11} w + D_{12} u$$
$$y = C_2 x + D_{21} w + D_{22} u$$

Now it's time to design a scheme, see Fig.3. Controller will have 2 inputs to cover also vertical level of the chassis zb. There might be a roughness on the road or the level of chassis is changed due to external conditions, both cases require certain actuating force.

The scheme contains $Hsens$ term, see Fig.4. It's MIMO system that reconstructs road level $z_{r,r}$ and chassis level $z_{b,r}$ from measured quantities $y_1 = z_b - z_r$ and $y_3 = z_b$". These signals are disturbed by noises n_1 a n_2. Disturbances are indeed included in the input of $Hsens$ but it's more convenient to consider them in the output. Unconnected output y_2 will be used to verify behaviour of the wheel on the road. Transfer function W_{int2} should be represented by system with astatism of the second order ($zb^{''}$ must be integrated

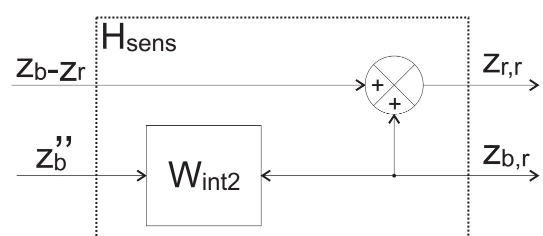

Figure 4. Realization of $Hsens$ term

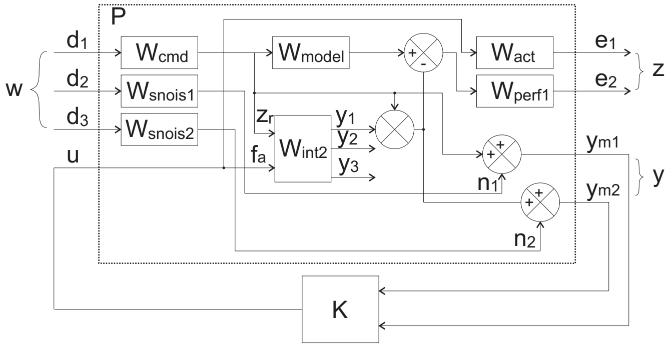

Figure 5. Block scheme of extended plant for H-∞ design

twice) but it will be approximated by the second order system with time constant of 100 s. This approach leads to zero steady state of reconstructed $z_{b,r}$ during long-term zero chassis acceleration.$W_{int2} = \frac{1}{s^2+0{,}02s+0{,}0001}$.

4 Extension of the model for H-∞ design

The plant G will be extended by weighting functions W, see Fig.5. There's a lot of ways to do that but however, the more external inputs w and error outputs there are, the more complicated is the choice of weighting functions and taking them into consideration during computations. Weighting functions are generally transfer functions which means that order of the system grows with its number. Thus modified system will be used for calculation of controller K. Input weighting signals W_{cmd}, W_{snois1} and W_{snois2} determine assumed form of input quantities, with complex transfer function $1(j\omega)$ and their outputs in appropriate physical units.

Input weighting signals W_{act}, W_{perf1} determine reciprocal value of expected form of output quantities, with their inputs in appropriate physical units and with unitary outputs.

If designed controller K doesn't meet the requirements the weighting functions must be further modified, that is to choose compromise among the requirements. Frequently used approach is multiplication the weighting function by a constant, which corresponds to a change of expected inputs and outputs. W_{cmd} means expected form of road level:

$$W_{cmd} = \frac{d_{r,\max}}{T_{pneu}s+1} = \frac{0,1}{0,01s+1}$$

where $d_{r,max} = 0,1m$ is a maximum step change of the road and $T_{pneu} = 0,01s$ is a time constant of raid on a roughness, corresponding average velocity of $20m/s$, roughness of $10cm$ and wheel diameter of $60cm$. W_{snois1},W_{snois2} are sensor noises (Gaussian expected), $W_{snois1}(s) = W_{snois1}(s) = 0,01$. W_{model} is desired (ideal) control transfer function after regulation. For this case a smooth step response is suitable.

$$W_{model} = \frac{1}{T_{mod}s^2 + 2\xi_{mod}T_{mod}s + 1} = \frac{1}{0,04s^2 + 0,2s + 1}$$

where T_{mod} = 0,2 s is steady-state time constant, ζ_{mod} = 1 is relative damper ($\zeta_{mod} < 1$ – regulation with overshoot, $\zeta_{mod} > 1$ – over-damped control process), W_{act} is reciprocal transfer function of actuating drive which is supposed to provide maximum force of 10 kN with zero frequency and works up to 50 rad/s. Realization time constant must be added in addition.

$$W_{act} = \frac{0,02s+1}{10^4\left(10^{-6}s+1\right)}$$

W_{perf1} is reciprocal value of expected variation from required control transfer function W_{model}:

$$W_{perf1} = \frac{0,2}{0,1s+0,0001}$$

State matrices for extended system P will be computed by Matlab (originally its the 4^{th}-order system plus 5 orders due to weighting functions). $Hsens$ was excluded of computation (due to 11^{th} order - with no optimal solution found) and ideal values of z_r a z_b of measured outputs were used provided that well-designed controller will work for this term too.

5 Simulation: connecting Simulink and Virtual Reality Toolbox

As inspiration for this project, one of the demos of Virtual Reality Toolbox, $vr\_octavia.mdl$, was used. Model used here in this project consists of wheels, chassis and road, parameters of which (positions, road profile) are directly controlled by Simulink inputs/outputs. Road profile z_r is predefined and wheel position z_w and chassis position z_b are visualized. By manual switch, it is possible to choose either or robust control. It is also possible to open a Simulink scope to see current course of these quantities. This is shown by Fig.6. Then, Fig.7. and Fig.8. show comparison for PID and robust control process for different masses $m_b = 250\ kg$ and $m_b = 350\ kg$. Fig.9. is comparison of control process, $m_b = 350\ kg$, for PID and H-∞ controllers operating during the simulation.

6 Virtual Reality model

The Virtual Reality Toolbox™ lets you view and interact with dynamic system simulations in a 3-D virtual

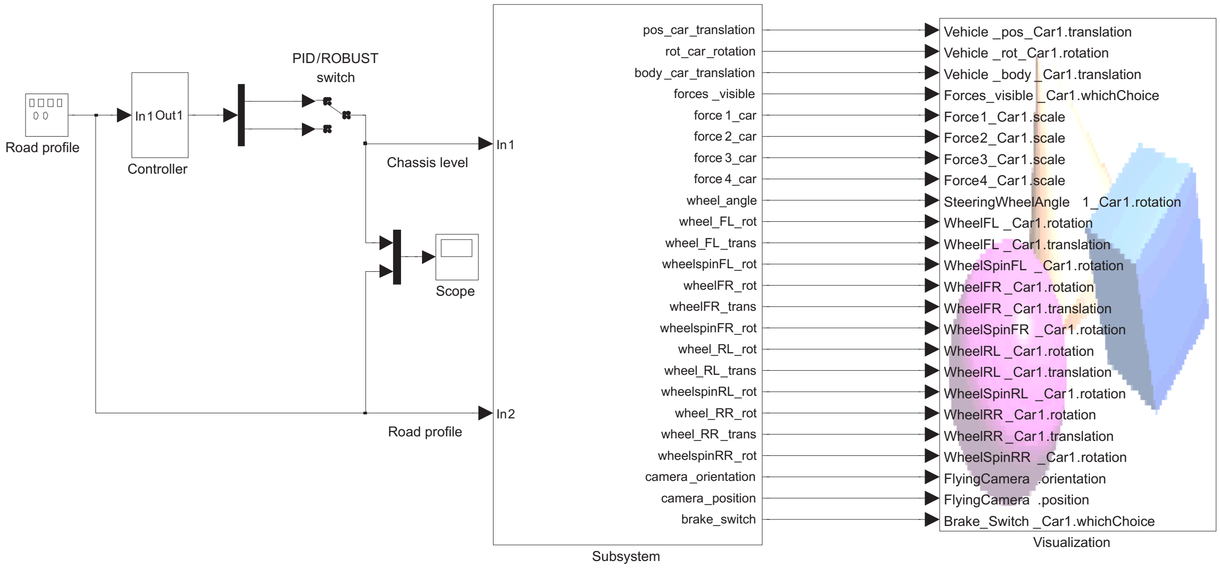

Figure 6. Simulink scheme

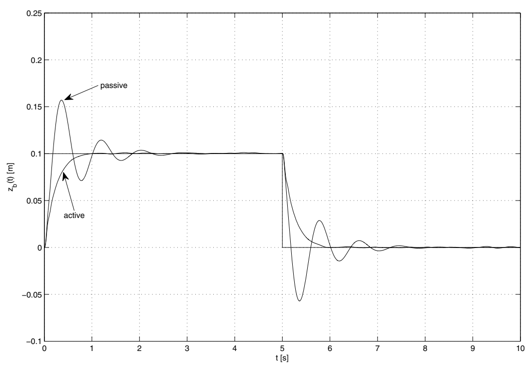

Figure 7. PID vs. H-∞, m_b=250 kg

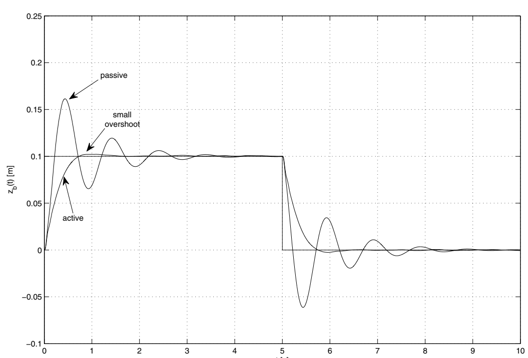

Figure 8. PID vs. H-∞, m_b=350 kg

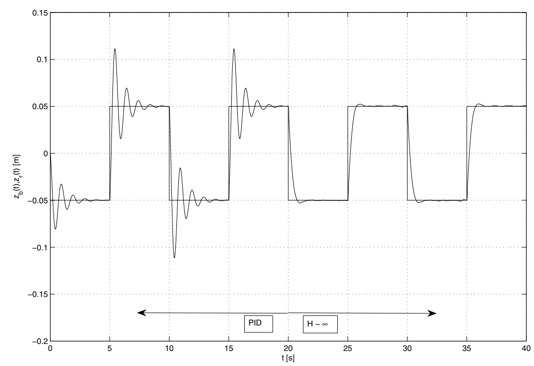

Figure 9. Quality comparison: m_b=350 kg, t=0-20s: PID control, 20-40s:Robust Control

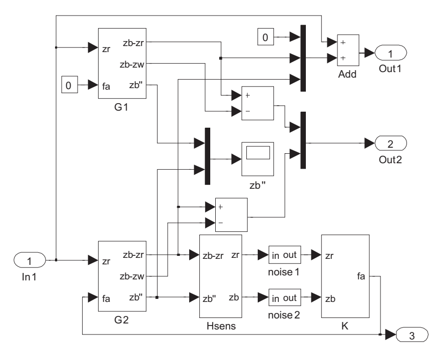

Figure 10. Controller Simulink Subsystem

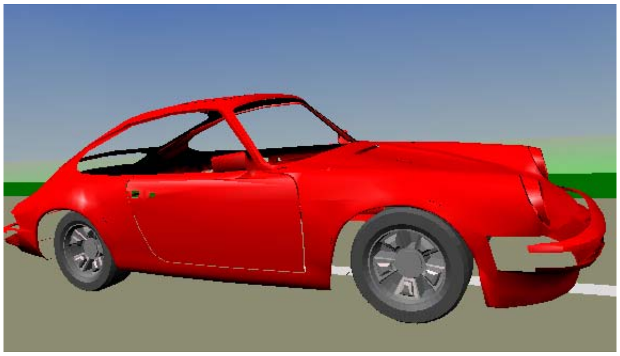

Figure 11. Model of a car in Virtual Reality Viewer

reality environment. The toolbox links MATLAB® and Simulink® with virtual reality graphics, enabling MATLAB or Simulink to control the position, rotation, and dimensions of the 3-D images defined in the virtual reality environment. The result is a presentation-quality 3-D animation. Through visualization, the Virtual Reality Toolbox product provides insight into the dynamic systems that you model in Simulink. You can use these viewing capabilities in automotive, aerospace, communication, biological, and mechanical applications.

7 Conclusions

Virtual Reality Toolbox has become a vivid education aid at the Department of Measurement and Control. It can be used in several subjects, namely Control Systems or Design and Realization of Controllers. Models can be viewed not only on local machine by internal viewer, but also on client workstations by means of web explorer.

Designed controller is a system of 9^{th} order. Each input of the controller asserts itself in a different way. This confirms that choice of two-state representation of a controller was correct. The results of the simulation show that ride comfort increased because the course of chassis level is smoother, with almost no overshoot, and the values of acceleration are smaller now. It also shows that chassis weight has almost no affect on operation of a controller. Stability of the wheel assembly is also better because the control process is shorter.

Simulation also verified the possibility of connection of this system into a real system or physical model because the model didn't rely on knowledge of state variables but it was able to reconstruct them from measured values.

Control process could be improved by considering other error inputs and weighting functions, for example direct penalization of ride comfort according the frequencies unpleasant for human. The position of the wheel can be also described in more details. However, experiments show that control process of such extended model has worse quality. Due to complexity of this topic a lot of effort must be invested to improve the model and to come closer to real operational conditions. There are many other ways in which the model can be described. Firstly, tire damping coefficient should be introduced into the model (analogous to c_s term between chassis and wheel assembly) and so the dynamics of tire itself. Secondly, non-linear model should be considered instead of linearized representation, including dynamics of actuator providing the force f_a should be covered. A half car model and full (4-wheels) model will be created in the next phase.

Using special toolboxes, concerning modern methods of design (rapid prototyping,HIL) the controller can be implemented on a chosen hardware. This methodology was tested for Motorola/Freescale HC12 microcontrollers using Real Time Workshop (compiling the Simulink schema into C code and direct loading into the chip). It is also planned to try the functionality on dSPACE systems – 1st class systems for control and simulation in real time which belongs to Matlab family, too.

8 Acknowledgment

This work was supported by the Ministry of Education of the Czech Republic under Project 1M0567.

References

[1] William R. Sherman, Alan Craig: *Understanding Virtual Reality: Interface, Application, and Design.* Morgan Kaufmann, 2003.

[2] Donald Bastow, Geoffrey Howard, John P. Whitehead: *Car Suspension and Handling.* SAE International, 2004.

[3] Tom Birch: *Automotive Suspension and Steering Systems.* CENGAGE Delmar Learning, 1998

[4] Katsuhiko Ogata: *Modern Control Engineering.* Prentice Hall, 2001.

[5] Richard C Dorf, Robert H. Bishop: *Modern Control Systems.* Prentice Hall, 2007.

[6] Ashish Tewari: *Modern Control Design With MATLAB and SIMULINK.* Wiley, 2002.

[7] Kemin Zhou, John C. Doyle: *Essentials of Robust Control.* Prentice Hall, 1997.

[8] Da-W Gu, P. Hr. Petkov, M. M. Konstantinov: *Robust Control Design with MATLAB.* Springer, 2005.