Comparison of Optimal, Robust and Model Predictive Control Structures for Quarter Car Active Suspension

1st Ceoca Ovidiu
Department of Automation
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
ceoca.ovidiu@gmail.com

***Abstract*—To ensure optimal performance of the active suspension system across various conditions and uncertainties, advanced control techniques such as H_∞, $\mu - Synthesis$ and Model Predictive Control (MPC) are employed. H_∞ control focuses on minimizing the worst-case peak gain from disturbances to outputs, ensuring robust performance even under significant uncertainties. $\mu - Synthesis$ also known as structured singular value synthesis, enhances this approach by addressing structured uncertainties, enabling the design of controllers robust against a wider range of variations in system parameters, such as actuator models. Complementing these methods, Model Predictive Control leverages real-time optimization and predictive capabilities to adapt dynamically to disturbances and changing conditions, further enhancing system adaptability and performance.**

***Index Terms*—Robust Control, Active Suspension, $\mu - Synthesis$, Quarter Car Model, Model Predictive Control**

I. Introduction

Active suspension systems have become increasingly important in the automotive industry due to their ability to significantly improve ride comfort and handling performance. One common approach to studying and designing such systems is through the use of a quarter-car model, which simplifies the complex dynamics of a full vehicle suspension to a one-dimensional system consisting of a sprung mass (representing the vehicle body), an unsprung mass (representing the wheel and tire assembly), and various springs and dampers to simulate the suspension components' behavior. This model allows engineers to focus on the key dynamics involved in suspension performance, including how the suspension responds to road disturbances and how it can be controlled to achieve optimal ride quality and stability. In an active suspension system, actuators are integrated to apply forces based on real-time measurements and control algorithms, enabling the suspension to actively adjust its characteristics to match driving conditions.

II. Physical Model

The quarter car model of the active suspension system is represented in Figure 1. The value m_b in kilograms represents the car chassis (body) and the value m_w in kilograms represents the wheel assembly. The spring and damper represent the passive spring and shock absorber placed between the car body and the wheel assembly. The spring models the compressibility of the pneumatic tire. The variables x_b, x_w, and r (all in meters) are body travel, wheel travel, and road disturbance, respectively. The force applied (in kilograms) between the body and wheel assembly is controlled by feedback and represents the active component of the suspension system.

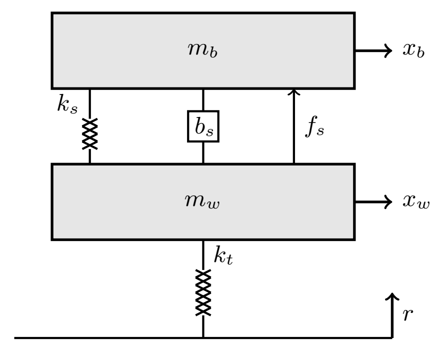

Fig. 1. Physical model of a quarter car active suspension

III. System

In Figure 2 the entire structure of the system is illustrated. There are two main signals that must be taken into account when designing the controller. The variable a_b represents the acceleration of the vehicle body and s_d is the movement of the suspension.

Based on these two signals, the controller should be able to adjust the hydraulic fluid force f_s, which represents the active part of the process. The controller alone cannot do that because the command from it is in electric form. For this matter, an actuator is added to the process that is responsible for adjusting the hydraulic force. The transfer function of the actuator is approximated to a first-order system represented in equation 1.

$$H_{act} = \frac{1}{0.01667s + 1} \tag{1}$$

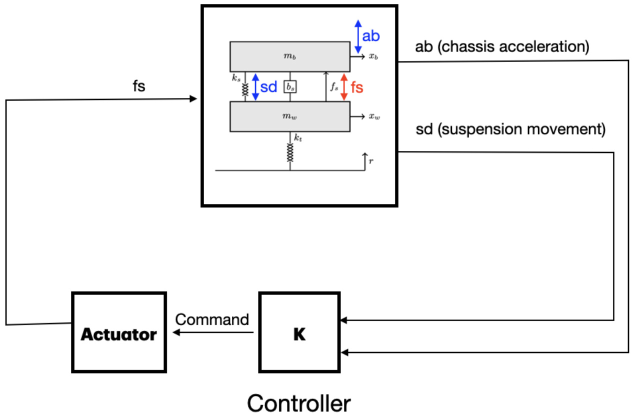

Fig. 2. The structure of the system

IV. Mathematical Model

The state-space model of the quarter car active suspension is represented by the 2, 3, 4 and 5 equations.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-k_s}{m_b} & \frac{-b_s}{m_b} & \frac{k_s}{m_b} & \frac{b_s}{m_b} \\ 0 & 0 & 0 & 1 \\ \frac{k_s}{m_w} & \frac{b_s}{m_w} & \frac{-k_s-k_t}{m_w} & \frac{-b_s}{m_w} \end{bmatrix} \tag{2}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & \frac{1e3}{m_b} \\ 0 & 0 \\ \frac{k_t}{m_w} & \frac{-1e3}{m_w} \end{bmatrix} \tag{3}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ \frac{-k_s}{m_b} & \frac{-b_s}{m_b} & \frac{k_s}{m_b} & \frac{b_s}{m_b} \end{bmatrix} \tag{4}$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{5}$$

The values used to create the state space model are described in equation 6.

$$\begin{aligned} m_b &= 300[Kg] \\ m_w &= 60[Kg] \\ b_s &= 1000[\frac{N}{m \cdot s}] \\ k_s &= 16000[\frac{N}{m}] \\ k_t &= 190000[\frac{N}{m}] \end{aligned} \tag{6}$$

The disturbance from the road is a signal with the amplitude of 0.05 (which is translated as a speed bump with the height of 5 cm) and it is illustrated in Figure 3.

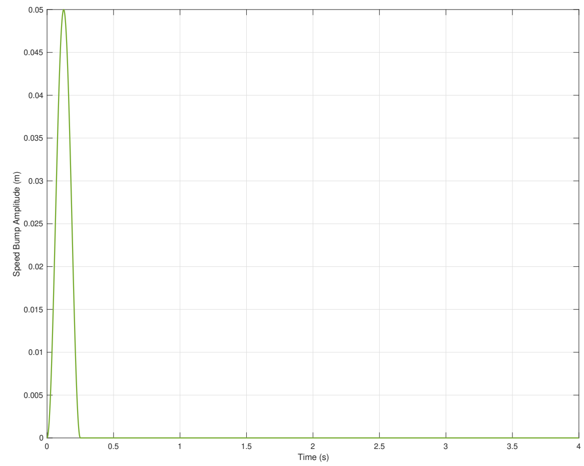

Fig. 3. Road Disturbance

V. H_∞ Controller Design

Firstly, the controller K has been designed using the H_∞ method in MATLAB with the help of the command *hinfsyn* which takes as parameters:

1) The computed system
2) The number of measurable signals (a_b and s_d)
3) The number of controlled signals (f_s)

The system responses for each mode (Standard, Comfort and Sport) and the open loop response are illustrated in Figures 4, 5, 6 and 7.

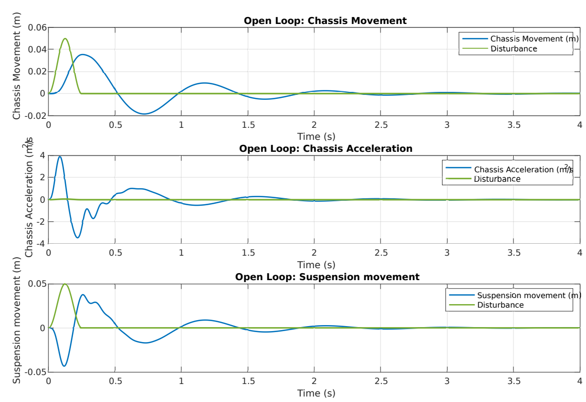

Fig. 4. Open loop response

It can be seen that by applying a controller, the overall motion of the chassis is reduced compared to the open loop response. The most aggressive response is from the *Sport* mode while the most convenient one, which responds very good to the disturbance is the *Comfort* one.

In terms of the Acceleration and the Suspension travel, the *Sport* mode reaches lower values because the suspension must be stiffer than the other modes, while the *Comfort* mode is the exact opposite. The *Standard* mode is somehow between them with sharp response to the disturbance, but not reaching so high values.

VI. $\mu - Synthesis$ Controller Design

All the simulations above are for a specific one case, while in practice the components of the car can vary from one to another. Moreover, in time, the quality of the parts reduces, but the controller should be able to compensate these kind of problems and deliver the same experience to the user.

In this case just a quarter of a car was taken into consideration, but the car has four active suspensions and also the actuator parameters from the transfer function can vary a lot. In order to design a controller to overcome these problems, a family of responses (50 in total) were simulated for each driving mode represented in Figures 10 using the H_∞.

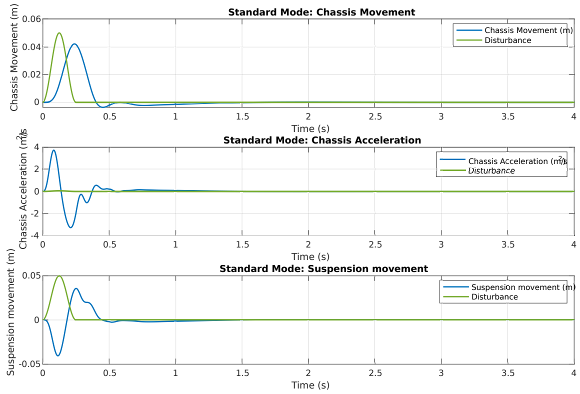

Fig. 5. Standard response

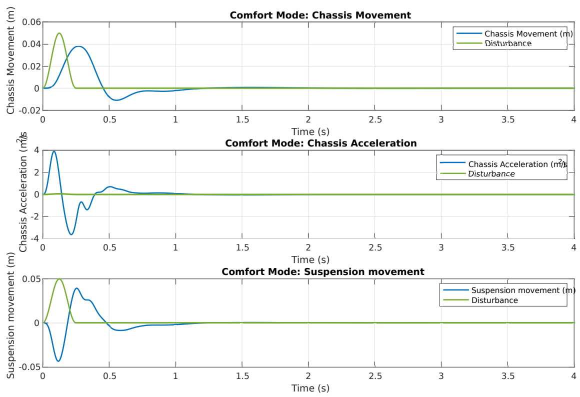

Fig. 6. Comfort response

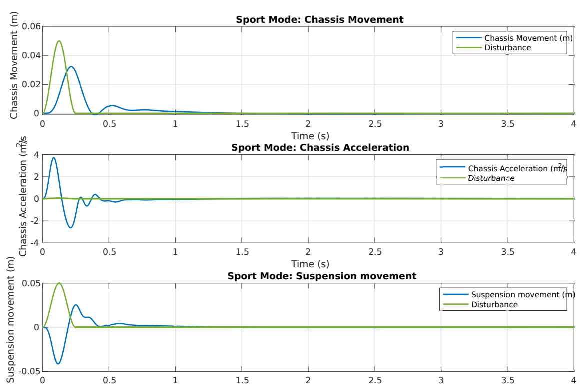

Fig. 7. Sport response

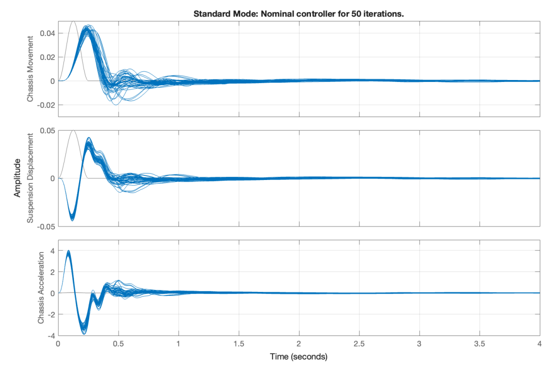

Fig. 8. Mulitple representation of H_∞ responses for Standard mode

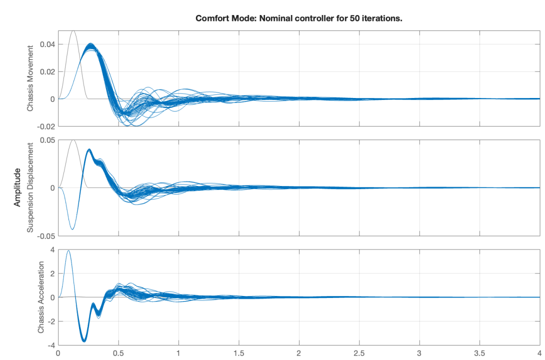

Fig. 9. Mulitple representation of H_∞ responses for Comfort mode

It can be seen that for all 50 iterations (which translates in real life as 50 possible suspension system for a car) the *Standard* and *Comfort* mode works really great. The sport mode is where the problems start to appear. For many iterations, the controller cannot handle the system, thus leading to the instability of the car.

To design the controller which can overcome the problems, the MATLAB command *musyn* was used and calculate the optimal controller taken into account the uncertainties from

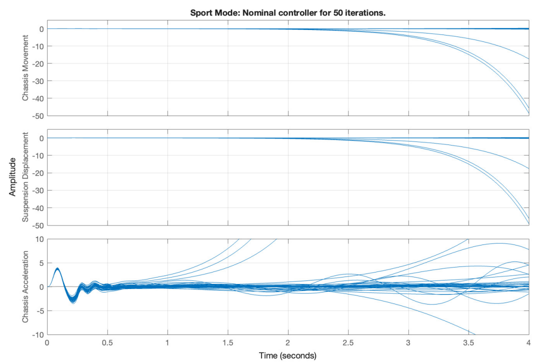

Fig. 10. Mulitple representation of H_∞ responses for Sport mode

the system. The uncertainties of the system can vary from the simple temperature changes to material fatigue and even errors in dynamics of the suspension. These uncertainties must be overcomed in order to assure the safety of the mechanism. The responses of the system with the $\mu - Synthesis$ controller are in Figure 11.

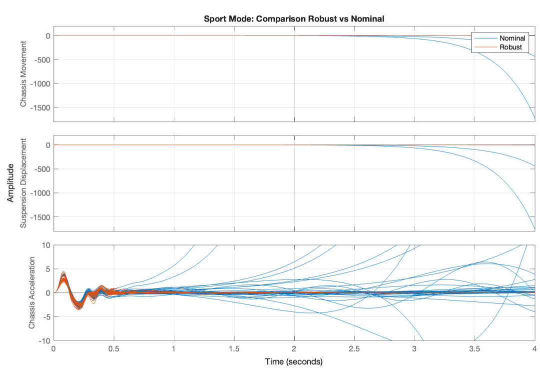

Fig. 11. System with $\mu - Synthesis$ controller for Sport mode

Now the responses of the system are all stable. The red signals are from the controller using the $\mu - Synthesis$ method and the blue ones are from the H_∞ controller.

VII. Model Predictive Control

As can be seen above, the controller using the $\mu - Synthesis$ method works fine for many cases, but the system response is quite slow for a sport car (0.5 seconds). This problem can be solved if the system would be able, based on different scenes on the road, to predict what should happen with the car. Fortunately, this can be achieved using special structure for predictive control.

First thing was to create a discrete model from the state space. The sample time used was $T_s = 0.01$. There are two important elements when it comes to predictive controller: prediction horizon and control horizon. The prediction horizon is set to $N_p = 20$ and the control one to $N_c = 10$. To create an MPC object in matlab, there is the dedicated command called mpc which takes as parameters: the system, T_s, N_p and N_c. To simulate the response of this object, there is the command sim. The simulation are presented in Figures 12, 13 and 14.

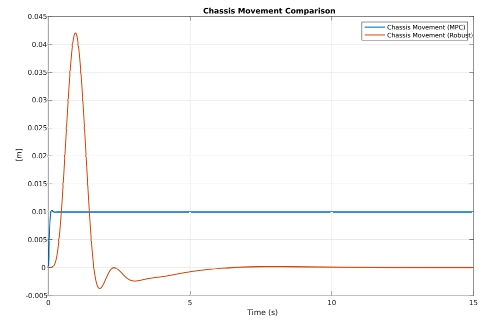

Fig. 12. Chassis movement using a predictive controller

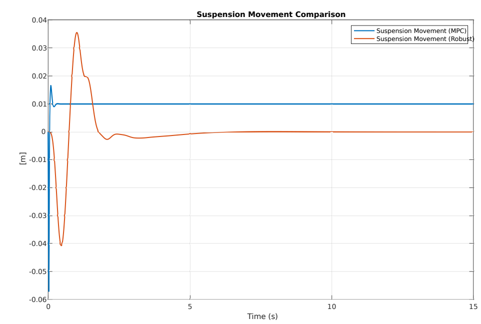

Fig. 13. Suspension movement using a predictive controller

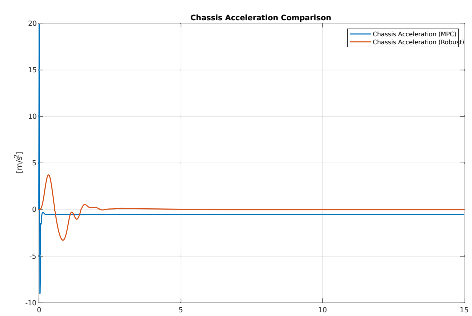

Fig. 14. Chassis acceleration using a predictive controller

With red there is the responses from the robust controller and with blue, the predictive one. The times are way faster using the predictive one than the robust one as can be seen in the Figures. The downside of this is that the acceleration of the chassis is much higher which translates as the car is keeping the suspension as stiff as possible to get these results. For a sport car, it is bearable, but when it comes to a day to day drive, it can be annoying. This can be adjusted from the parameters mentioned above to get the best out of these two, the benefits of the robust controller and the times from the predictive one.