System Approach to Vehicle Suspension System Control in CAE Environment

Vladimir Popovic[1,*] - Branko Vasic[1] - Milos Petrovic[2] - Sasa Mitic[1]
[1] University of Belgrade, Faculty of Mechanical Engineering, Serbia
[2] Institute for Research and Design in Commerce & Industry, Serbia

In recent years, motor vehicles industry has shown a tendency of replacing electromechanical components by mechatronic systems with intelligent and autonomous properties. The integration of hardware components and implementation of advance control function characterize this replacement. In this paper, we have applied the system approach and system engineering methods in the initial phase of vehicle active suspension development. An emphasis has been placed upon the interrelations between computer-aided simulation and other elements of the development process. The benefits of application of active suspension simulation are numerous: reduction of time to market, the new and improved functions of mechatronic components/devices, as well as the increased system reliability. In suspension model development, we used CAD/CAE tools, as well as the multipurpose simulation programs. For simulation, we used the one-quarter vehicle model. The modelling was carried out through the state-space equation, after which we designed the controller for our system. During this, we considered only the digital systems of automatic regulation.

0 INTRODUCTION

Considerable effort aiming at reducing the cost, the ever increasing expectations of users when it comes to reliability, the greater complexity of modern technical systems, and other requests, bring us to the conclusion that the approach based on system engineering principles, remains the only way to stay competitive on the market. Intensifying the integration of electronic components into areas which used to be strictly mechanical and the degree of change within technologies, all result in increased demands for dealing with the problems on the basis of what is offered by system engineering. In recent years, motor vehicles industry has started developing an unusual trend - the mechatronic systems with intelligent and autonomous abilities. Development of integrated mechatronic systems should, in its own way, play the key role in motor vehicles industry [1]. Mechatronics is being more and more accepted as the design methodology for all motor vehicle systems. Verification of proper functionality through simulation can improve reliability and reduce component design time. Replacement of simple electro-mechanical components by "intelligent" mechatronic systems is characterised by two aspects: integration of hardware components and implementation of advanced control functions (functional and algorithm integration). Hence, mechanical components are operated through basic sensor feedback of low level and intelligent processing of advanced level information.

The goal in active suspension control research is to improve the ride performance, generally quantified by sprung mass acceleration, while maintaining an acceptable level of suspension stroke and tyre deflection as packaging and handling measures. Ride comfort, road holding ability and suspension deflection are the three main performance criteria in any vehicle suspension design [2]. Finally, all control concepts aim at introducing additional forces to the suspension system to reduce roll, pitch, and stroke movements, as well as body and wheel vibrations. Therefore, the basic control approach is similar for all actuation systems. Then, the physical structure of the considered system determines the further controller design and also directly limits the achievable system's performance [3].

*Corr. Author's Address: University of Belgrade, Faculty of Mechanical Engineering, Kraljice Marije 16, 11120 Belgrade, Serbia, vpopovic@mas.bg.ac.rs

1 DESIGN DEVELOPMENT PROCESS

The design procedure, which is to be carried out in the case of mechatronic systems, is very demanding - it is necessary that system engineering, within the field of mechanics, electronics and computer science, forms a completely integrated system. Hence, intelligent testing technologies, supported by CAE (Computer Aided Engineering), which provide numerical simulation models, are to be employed during component development and their qualification. An emphasis should be placed upon the interaction between computer aided system simulation and experimental testing techniques through intelligent information processing. CAE is a technology that enables computer analysis of the design, created within CAD (Computer Aided Design) technology. By the application of CAE technology it is now possible to have a much better linking of design, testing and design improvement, which are the development phases that have, so far, been almost separate. Fig. 1 shows a diagram of design development process [4]. After theoretical modelling and experimental system identification, the future design steps are control system analysis and model based controller design. The demands for efficient optimisation and testing include:

• Software development environment with accurate, firm connections among different CAE tools, such as CAD, FEM and MBS programs, and CACE (Computer Aided Control Engineering) tools. An example of such an environment is shown in Fig. 2 [5]. On the basis of this diagram, we can determine the role of Matlab (and Simulink) in the framework of the complete process of modelling and simulation of the new technical system.

• Experimental, high performance platform which, in combination with software development environment, completely automates the implementation process from off-line simulation to engaging complex controllers with real time operation.

Interactive testing of mechatronic components, using advanced hardware simulators in feedback, plays an important role. The idea is to build mechatronic components into a virtual environment in which vehicle movement, external load and the surrounding mechanical systems, are simulated in real time. A complex model of vertical vehicle dynamics has been developed using CAE tools and multi-purpose simulation programs. On the basis of reduced models, a multivariable, powerful control system should be designed for active suspension control. Fig. 3 [4], [5] shows the corresponding ¼ vehicle model with a passive spring that bears static load, which is to be used for suspension system simulation.
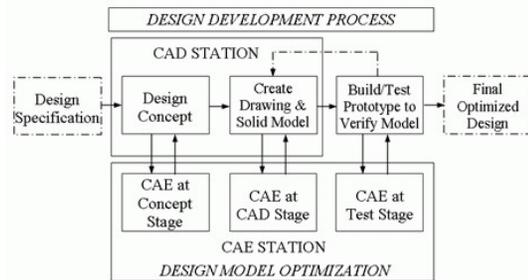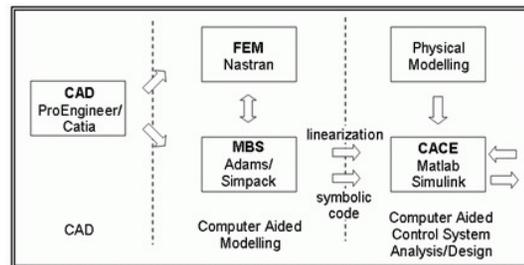

Fig. 1. *Design development process*

Fig. 2. *Software development environment with connections among different CAE tools*

2 ACTIVE SUSPENSION SYSTEM MODELLING

Various approaches have been proposed to improve the performance of active suspension designs, such as linear optimal control, fuzzy logic and neural network control, adaptive control, H_∞ control, nonlinear control, LQG control, skyhook control algorithm, etc [2] to [8]. Also, many approaches are presented to deal with the multi-objective requirement of vehicle suspensions. In particular, H_∞ control of active suspensions is intensively discussed in the context of robustness and disturbance attenuation [2]. An interesting solution is a fuzzy sliding mode controller to

control an active suspension system and evaluate its control performance [9] to [12]. The fuzzy sliding mode controller employed the error of the sprung mass position and the error change to establish a sliding surface, and then introduced the sliding surface and the change of the sliding surface as input variables of a traditional fuzzy controller in controlling the suspension system [13]. One of the possible solutions is vibration control of a vehicle active suspension system using a new robust neural network control system [14]. It consists of a robust feedback controller and feedforward neural network predictive controller.
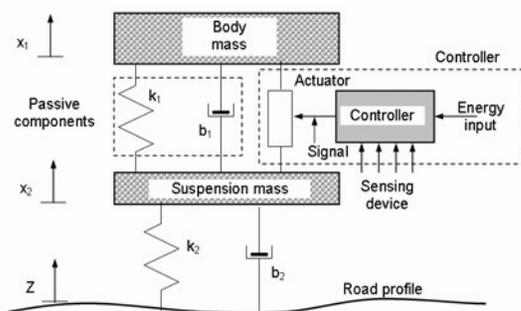
Fig. 3. *1/4 vehicle model*

In the design of the suspension system, the following is used: ¼ vehicle model [2] to [8], [11] and [13], ½ vehicle model [15] and full vehicle model [14] and [16]. We used ¼ vehicle model (Fig. 3) to simplify the problem to a one-dimensional spring-damper system. The vehicle represents a complex oscillatory system with a great number of freedom degrees. Vehicle oscillations are caused mainly by road disturbance. In reviewing vehicle oscillations we adopted the following starting points:

- the vehicle is in the rectilinear motion at a constant speed;
- the wheels are always in contact with the road and that is a one-spot contact;
- road disturbance is the same on the left and the right wheel and the car is symmetrical relative to the longitudinal axis;
- mass distribution coefficient is approximately 1.

Although the real characteristics of the suspension system (k_1, b_1) are non-linear, we adopted constant values for them in this paper which, to a certain extent, enable the linearization

of the model. Also, the step function that we used for simulation has certain limitations since if the tyre of the vehicle is excited by the step function, the tyre will jump and the contact with the driving surface will be lost. Furthermore, when compressed, damping of the suspension damper is many times smaller than damping when the damper is extended. These limitations will be further discussed in the conclusion. The data that we used during the simulation refer to the bus. The simulation was performed using Matlab interactive environment. The designations in Fig. 3 have the following meanings [4] and [5]:

- body mass (m_1) = 2250 kg;
- suspension mass (m_2) = 290 kg;
- spring constant of suspension system (k_1) = 72000 N/m;
- spring constant of wheel and tyre (k_2) = 450000 N/m;
- damping constant of suspension system (b_1) = 315 Ns/m;
- damping constant of wheel and tyre (b_2) = 13500 Ns/m;
- control force (F_a) = force from the controller we are going to design.

When the vehicle is experiencing any road disturbance, the body should not have large oscillations, and the oscillations should dissipate quickly. This, at the same time, is our principal task. Since the distance x_1-Z is very difficult to measure, and the deformation of the tyre x_2-Z is negligible, we will use the distance x_1-x_2 instead of x_1-Z as the output in our problem. The road disturbance Z will be simulated by a step input. This step could represent the vehicle coming out of a pothole. We wish to design a feedback controller so that the output x_1-x_2 has an overshoot less than 5% and a settling time shorter than 5 sec. Dynamic properties of the system can, in time range, be most suitably defined by the values of parametres that determine the system response (overshoot and settling time) [17].

2.1 Model of the System in State-Space

The assessment of the quality of automatic regulation of a system behaviour essentially boils down to estimating the error between a predetermined value and the real value of the controlled variable. The knowing of this error at

any point would give complete information on the features of the observed system. Due to the diversity of the laws of the system input change, which might occur in the normal work regime, such an approach, based on estimating the current values of error, is inappropriate speaking from the aspect of practice. Therefore, it is preferable to make an estimation of the relevant system characteristics on the basis of the features it manifests while being disturbed by the typical input signals [17]. On the basis of Fig. 3 and Newton's Law, the following dynamic equations, which represent the mathematical model of the dynamic system [4] and [5] are obtained:

$$m_1 \ddot{x}_1 = -b_1(\dot{x}_1 - \dot{x}_2) - k_1(x_1 - x_2) + F_a \qquad (1)$$

$$m_2 \ddot{x}_2 = b_1(\dot{x}_1 - \dot{x}_2) + k_1(x_1 - x_2) + \\ + b_2(\dot{Z} - \dot{x}_2) + k_2(Z - x_2) - F_a. \qquad (2)$$

When addressing new ways for system analysis and synthesis that avoid the problem of solving algebraic and differential equations, the system analysis and synthesis in state-space mainly come to mind. Representation of the system in the form of state-space equation is much easily derived from differential equations than by using Laplace transformations. To be a valid state-space representation, the derivative of all states must be in terms of inputs and the states themselves. Now, let us choose the states that we shall be using. Firstly, let us divide the Eqs. (1) and (2) by m_1 and m_2, respectively and introduce the substitute $Y_1 = x_1 - x_2$. Note that $\dot{Z}$ appears in the Eq. for $\ddot{x}_2$:

$$\ddot{x}_1 = -\frac{b_1}{m_1}\dot{Y}_1 - \frac{k_1}{m_1}Y_1 + \frac{F_a}{m_1}, \qquad (3)$$

$$\ddot{x}_2 = \frac{b_1}{m_2}\dot{Y}_1 + \frac{k_1}{m_2}Y_1 + \frac{b_2}{m_2}(\dot{Z} - \dot{x}_2) + \\ + \frac{k_2}{m_2}(Z - x_2) - \frac{F_a}{m_2}. \qquad (4)$$

The first state-space equation will be x_1. Since no derivatives of the input appear in the equation for $\ddot{x}_1$, we choose $\dot{x}_1$ for the second state. Then, we choose the third state as the difference between x_1 and x_2. After doing some algebra, we will determine what the fourth state

should be. Therefore, we subtract Eq. (4) from Eq. (3) to get an expression for $\ddot{Y}_1$:

$$\ddot{x}_1 - \ddot{x}_2 = \ddot{Y}_1 = -(\frac{b_1}{m_1} + \frac{b_1}{m_2})\dot{Y}_1 - (\frac{k_1}{m_1} + \frac{k_1}{m_2})Y_1 - \\ - \frac{b_2}{m_2}(\dot{Z} - \dot{x}_2) - \frac{k_2}{m_2}(Z - x_2) + F_a(\frac{1}{m_1} + \frac{1}{m_2}). \qquad (5)$$

Since we cannot use second derivatives in the state-space representation, we integrate this Eq. to get $\dot{Y}_1$:

$$\dot{Y}_1 = -(\frac{b_1}{m_1} + \frac{b_1}{m_2})Y_1 - \frac{b_2}{m_2}(Z - x_2) + \\ + \int(-(\frac{k_1}{m_1} + \frac{k_1}{m_2})Y_1 - \frac{k_2}{m_2}(Z - x_2) + \\ + F_a(\frac{1}{m_1} + \frac{1}{m_2}))dt. \qquad (6)$$

No derivatives of the input appear in this equation, and $\dot{Y}_1$ is expressed in terms of states and inputs only, except for the integral. Let us call the integral Y_2. Assuming that $x_2 = x_1 - Y_1$, from the Eq. (6) we get the state-space equation for Y_1:

$$\dot{Y}_1 = -(\frac{b_1}{m_1} + \frac{b_1}{m_2})Y_1 - \frac{b_2}{m_2}(Z - x_1 + Y_1) + Y_2. \qquad (7)$$

Then we shall substitute the derivative of Y_1 into the Eq. (3) of the derivative of x_1:

$$\ddot{x}_1 = -\frac{b_1 b_2}{m_1 m_2}x_1 + (\frac{b_1}{m_1}(\frac{b_1}{m_1} + \frac{b_1}{m_2} + \frac{b_2}{m_2}) - \frac{k_1}{m_1})Y_1 + \\ + \frac{b_1 b_2}{m_1 m_2}Z + \frac{F_a}{m_1} - \frac{b_1}{m_1}Y_2. \qquad (8)$$
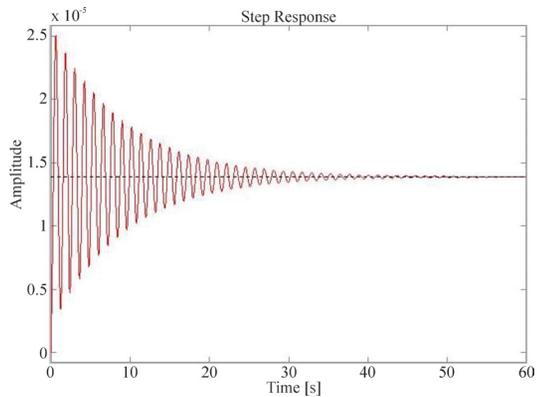

Fig. 4. *Open-loop response for the unit step actuated force*

The state variables are x_1, $\dot{x}_1$, Y_1 and Y_2. The matrix from the above Eq. is:

$$\begin{bmatrix} \dot{x}_1 \\ \ddot{x}_1 \\ \dot{Y}_1 \\ \dot{Y}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\dfrac{b_1 b_2}{m_1 m_2} & 0 & \dfrac{b_1}{m_1}\left(\dfrac{b_1}{m_1}+\dfrac{b_1}{m_2}+\dfrac{b_2}{m_2}\right)-\dfrac{k_1}{m_1} & -\dfrac{b_1}{m_1} \\ \dfrac{b_2}{m_2} & 0 & -\left(\dfrac{b_1}{m_1}+\dfrac{b_1}{m_2}+\dfrac{b_2}{m_2}\right) & 1 \\ \dfrac{k_2}{m_2} & 0 & -\left(\dfrac{k_1}{m_1}+\dfrac{k_1}{m_2}+\dfrac{k_2}{m_2}\right) & 0 \end{bmatrix} * \begin{bmatrix} x_1 \\ \dot{x}_1 \\ Y_1 \\ Y_2 \end{bmatrix} + $$

$$+ \begin{bmatrix} 0 & 0 \\ \dfrac{1}{m_1} & \dfrac{b_1 b_2}{m_1 m_2} \\ 0 & -\dfrac{b_2}{m_2} \\ \dfrac{1}{m_1}+\dfrac{1}{m_2} & -\dfrac{k_2}{m_2} \end{bmatrix} \begin{bmatrix} F_a \\ Z \end{bmatrix}, \qquad (9)$$

$$Y = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ \dot{x}_1 \\ Y_1 \\ Y_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} F_a \\ Z \end{bmatrix}. \qquad (10)$$

We can put the above state-space Eqs. (9) and (10) into Matlab by defining the four matrices of the standard state-space equation: $\dot{X}=AX+BZ$; $Y=CX+DZ$. Thus, a new *m*-file, formed on the basis of state-space equation has been created. By adding *'step(A,B,C,D,1)'* command into the m-file and its activating, in Matlab command window an open-loop response for the unit step actuated force is obtained. Generally, new functions to the Matlab vocabulary are added by expressing them in terms of existing functions. The existing commands and functions that compose the new function reside in a text file called an *m*-file [18]. *M*-files can be either scripts or functions. Scripts are simply files containing a sequence of Matlab statements. Functions make use of their own local variables and accept input arguments. The name of an *m*-file begins with an alphabetic character and has a filename extension of ".m".

Fig. 4 shows that the system is underdumped. People sitting on the bus will feel a small amount of oscillation and the steady-state error is about 0.01 mm. However, the bus needs unacceptably long time to reach the steady state ‒ the settling time is rather long. The solution to this problem lies in including a feedback controller into the block diagram of the system. By the term controller we also imply

the controller and the actuator. By adding the command *'step(A,0.1*B,C,D,2)'* into the *m*-file the open-loop response to 0.1 m step disturbance is obtained.
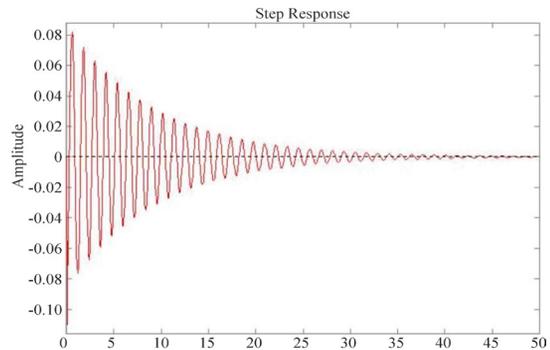

Fig. 5. *Open-loop response to 0.1 m step disturbance*

Fig. 5 shows that, when the bus passes a 10 cm high bump on the road, the bus body will oscillate for an unacceptably long time ($\approx$ 50 sec), and with a much larger amplitude than the initial impact. The big overshoot and the slow settling time will cause damage to the suspension system. As already stated above, the solution to this problem is to add a feedback controller into the system to improve performance. The block diagram of the thus obtained closed-loop system is presented in Fig. 6.
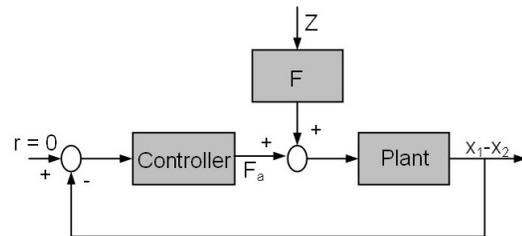

Fig. 6. *Closed-loop system block diagram*

3 SYNTHESIS OF ACTIVE SUSPENSION DIGITAL SYSTEM

Apart from the PID algorithm, which was used for controlling an active vehicle suspension system, many modern solutions for control systems and actuators themselves are found in the literature:

- H_∞ control with actuator time delay [2];

- Impedance control for the vehicle suspension system by electro hydraulic actuator – the electro hydraulic actuator comprises a servo-valve and a hydraulic cylinder [6];
- Hybrid control technique applied to a vehicle active suspension system using skyhook and adaptive neuro active force control [8]. The overall control system essentially comprises four feedback control loops; the innermost proportional-integral control loop for the force tracking of the pneumatic actuator, the intermediate skyhook and active force control control loops for the compensation of the disturbances and the outermost PID control loop for the computation of the optimum target/ commanded force. Pneumatic actuators have the advantage of low cost, a high power to weight ratio, ease of maintenance and a readily available and cheap power source;
- Different kinds of rotational actuators, electric levelling actuators, electromechanical actuators and electric damper actuators [7];
- The optimization technique used for the optimization of the controller parameters and the spring rates is based on a genetic algorithm [7]. The main advantage of this method and all other non-gradient based optimization methods is that it does not need the derivatives of the objective function, which are difficult to calculate from a numeric simulation model;
- Knowledge-based fuzzy logic controllers, variable structure controllers, backstepping controllers [16];
- Possible solutions are dampers taking advantage of the electro-rheological or magneto-rheological property of liquids. Both principles are based on the alteration of the damping medium's viscosity depending on an applied electric or magnetic field. These systems are mainly applied with semi-active vehicle suspensions [3] and [19];
- Somewhat older solutions are hydraulic, hydro-pneumatic, electro-mechanical and pneumatic systems. Most of these systems also comprise passive elements such as conventional springs and dampers or hydro-pneumatic damping and springing components involved in the actuation system [3];

- In the integrated intelligence type implementation, actuator intelligence and actuation are combined into a single material. A piezoceramic actuator is a typical example of this type [20].

Actuators may have undesirable static and dynamic characteristics introducing many problems for the PID control loop. For instance, limit cycles caused by actuator backlash cannot be removed by adjusting PID parameters. Therefore, a nonlinearity compensation algorithm using an inverse model and a simple position controller are implemented in order to maintain the actuator characteristics to be as linear as possible within the attainable actuation range [20]. The PID controller has been most widely employed because of its simple structure and the effective use in industry [21] and [22]. Despite many advances for the PID controller, this structure has constant gain parameters and is not good for decreasing velocity control error [14]. For the purpose of our paper, the simplest conventional PID controllers are obtained through an easy discretization of analogous equivalents. By the application of z-transformation, the function of incremental PID controller discrete transfer in the following form is obtained:

$$D_p(z) = \frac{U(z)}{E(z)} = K\left[1 + \frac{T/T_I}{1-z^{-1}} + \frac{T_D}{T}(1-z^{-1})\right]$$

or   (11)

$$D_p(z) = K_P + \frac{K_I}{1-z^{-1}} + K_D(1-z^{-1}),$$

where $K_P = K$, $K_i = KT/T_i$ and $K_D = KT_D/T$ are known as P-, I- D-action factors, respectively (K is the factor of gain; T_I and T_D are time constants of integral and differential action).

The structure of the system with a digital PID controller is given in Fig. 7 [17]. Note that without the block with broken lines in Fig. 7a, the regulation contour contains an incremental PID controller. With this block, however, this contour becomes equivalent to the contour in Fig. 7b, which contains a positional PID regulation law. The positional type of digital law seems natural; however, the incremental kind is more readily used. Incremental algorithm must be used if it is immediately connected to the executive organ of the stepping motor type, and it is also convenient

hydraulic actuators where the delays are taken by the actuators to build up the required control force [2]. Although the delay time may be short, it can nevertheless limit the control performance or even cause the instability of the system when the delay appears in the feedback loop.
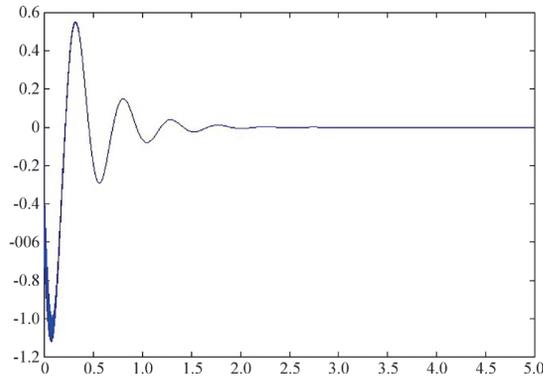

Fig. 8. *Closed-loop response to 0.1 m step disturbance*

3.1 Adding an Integrator

The next step is adding an integrator into the system, so that the system response in stationary state is zero. We will add this integrator in series with the plant. This will result in adding of the other system states. We shall add an integrator by representing it in state-space and by using the *'series'* command. This command takes A, B, C and D matrices of the two systems, links them in a serial connection as arguments and brings us back a new set of A, B, C and D matrices. The integrator in discrete state-space may be represented in any time T in the following way:

$$x(k+1) = x(k) + Tu(k) \quad \text{and}$$

$$y(k) = x(k) + \frac{T}{2}u(k). \tag{12}$$

To do this, let us add the following command into the *m*-file:

$$Ai = 1; \quad Bi = 1; \quad Ci = T; \quad Di = T/2;$$
$$[Ada\ Bda\ Cda\ Dda] =$$
$$= series(Ad,Bd,Cd,Dd,Ai,Bi,Ci,Di).$$

Matlab will respond to us with a new set of matrices, which are the result of integrator action, with matrix dimension 5 instead of the earlier dimension 4.

$$Ada = \begin{bmatrix} 1.0000 & 0 & 0 & 1.0000 & 0 \\ 0 & 1.0000 & 0.0005 & 0 & 0 \\ 0 & -0.0034 & 1.0000 & -0.0125 & -0.0001 \\ 0 & 0.0232 & 0 & 0.9762 & 0.0005 \\ 0 & 0.7652 & 0.0002 & -0.9051 & 0.9998 \end{bmatrix},$$

$$Bda = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0.0034 \\ 0 & -0.0232 \\ 0 & -0.7652 \end{bmatrix},$$

$$Cda = 1.0e\text{-}003 * (0.5000 \quad 0 \quad 0 \quad 0.2500 \quad 0),$$

$$Dda = 0 \quad 0.$$

Unfortunately, the output of this equation is a new integral value. The matrix output Cda must be changed, in which case we get:

$$Cda=[Cd\ 0]\ Cda = \ 0 \ 0 \ 1 \ 0 \ 0.$$

The controller structure is similar to that of the state-space controller with continual systems. We shall now use the *'place'* command, so as to calculate the gain matrix K, which will provide the desired poles of the closed-loop regulation system. First, we should decide where to position the closed-loop poles. Since we need to determine the position of all the five closed-loop poles, we must be very selective when it comes to this problem. To be more precise, we can define the poles so that they cancel all the system zeros, and at the same time give us the desired response. Firstly, we should track all the zeros by converting digital state-space equations into the transfer function, and then by determining the square root of the numerator. We shall use the *'ss2tf'* command which takes state-space matrices and the chosen input as arguments, and gives us the transfer function of the numerator and denominator as the output. Let us add the following code into the *m*-file:

$$[num,den]=ss2tf\ (Ad,\ Bd,\ Cd,\ Dd,\ 1);$$
$$zeros=roots(num)$$

The following response is obtained:

$$zeros = \ 0.9987 + 0.0065i$$
$$0.9987 - 0.0065i$$
$$-0.9929 \ .$$

We shall select these three zeros as three out of five desired poles of the closed-loop system. One of the remaining two will be selected at 0.9992, for it is the pole location after approximately 10000 sampling times (or 5 sec). The last pole will be selected with $z = 0.2$, for this is quick enough to be neglected. Let us add the following code into our *m*-file:

```
p1=.97+.13i;      p2=.97-.13i;
p3=-.87;
p1=zeros(1);      p2=zeros(2);
p3=zeros(3);
p4=.9992;         p5=.5;
```

K=place(Ada, Bda*[1;0], [p1 p2 p3 p4 p5])

Matlab gives us the following response:
place: ndigits = 15
$K = 1.0e+008 *$
* (0.0082 0.3458 0.0054 9.8881 0.0096) .

We shall use *'dstep'* command to simulate the closed-loop response. Since multiplying the state-space vector by K in our controller, gives one signal only, it is necessary to add a series of zeros to K vector by multiplying it by $[1\ 0]^T$. We shall perform the simulation by the negative value of the step disturbance -0.1 m, in order to get the positive value of the deflection for practical reasons. Let us add the following code:

```
yout=dstep(Ada-Bda*[1 0]'*K,
-.1*Bda, Cda, -.1*Dda, 2, 10001);
t=0:.0005:5;      stairs(t,yout) .
```

Fig. 8 shows the closed-loop response to 0.1 m step disturbance. The overshoot is less than 5 mm, and the settling time is not more than 5 sec.

4 CONCLUSION

Over the past few years, active suspension system has seen increasing application. The simulation of this system in the initial development phase is of a multiple importance: i) time is gained ─ by not having to wait for exploitation results; ii) it is cost-effective ─ an expensive model in the lab does not need to be designed and complex laboratory examinations need not be performed; iii) suspension systems of almost all categories and types of motor vehicles can be simulated, which would, be hard to achieve in laboratory conditions. Before the practical implementation of the active suspension system, it is recommended to simulate the obtained analytical solutions on the computer in order to examine the range of change of regulative action and the response of the regulated variable to a typical input signal, and thus confirm the possibility of the system physical realization, bearing in mind the actuator limitations, as well as check whether other possible limitations within the object of control as a whole have been reached. The above statements by no means suggest that laboratory testing is no longer necessary, but that it can be carried out in a shorter period of time and with reduced cost [24]. The key challenge associated with active suspension and their actuators is the size, weight and energy consumption required to achieve acceptable performance. For this reason, the physical properties of the actuator should be included in the optimization problem. In practice, actuator faults are quite often the largest source of control system degradation. Suspension system modelling was performed on ¼ vehicle model, using the Matlab interactive environment and by the state-space equation. The step input is the unit step function, that is, a certain value of road disturbance. It has been concluded that after encountering any kind of obstacle, the settling time and overshoot of the vehicle, are too long, and that a controller must be introduced into the suspension system. We designed a digital controller by the pole-placement method, which makes only one of the possible solutions. The presented dynamic model is only a very rough representation of the true dynamic system, which is applicable only in the early concept design phases of the research and development process. It is expected that, in the near future, this design and the above-mentioned modifications and the necessary improvements, would be used in designing such systems in the motor vehicles industry of our country.

5 ACKNOWLEDGEMENT

6 REFERENCES

[1] Bolton, W. (2008). *Mechatronics - electronic control systems in mechanical and electrical engineering*, 4th ed. Pearson Prentice Hall.

[2] Du, H., Zhang, N. (2007). H_∞ control of active vehicle suspensions with actuator time delay. *Journal of Sound and Vibration*, vol. 301, p. 236-252.

[3] Fischer, D., Isermann, R. (2004). Mechatronic semi-active and active vehicle suspensions. *Control Engineering Practice*, vol. 12, p. 1353-1367.

[4] Popovic, V., Jankovic, D., Vasic, B. (2000). Design and simulation of active suspension system by using Matlab. *Proceedings of FISITA World Automotive Congress,* paper F2000G269, Seoul.

[5] Popovic, V., Vasic, B., Jankovic, D. (2002). Development of semi-active suspension system in CAE environment. *Proceedings of FISITA World Automotive Congress,* paper F02V062, Helsinki.

[6] Fateh, M.M., Alavi, S.S. (2009). Impedance control of an active suspension system. *Mechatronics*, vol. 19, p. 134-140.

[7] Jonasson, M., Roos, F. (2008). Design and evaluation of an active electromechanical wheel suspension system. *Mechatronics*, vol. 18, p. 218-230.

[8] Priyandoko, G., Mailah, M., Jamaluddin, H. (2009). Vehicle active suspension system using skyhook adaptive neuro active force control. *Mechanical Systems and Signal Processing*, vol. 23, p. 855-868.

[9] Huang, S.J., Chen, H.Y. (2006). Adaptive sliding controller with self-tuning fuzzy compensation for vehicle suspension control. *Mechatronics*, vol. 16, p. 607-622.

[10] Sharkawy, A.B. (2005). Fuzzy and adaptive fuzzy control for the automobiles' active suspension system. *Vehicle System Dynamics*, vol. 43, no. 11, p. 795-806.

[11] Taskin, Y., Hacioglu, Y., Yagiz, N. (2007). The use of fuzzy-logic control to improve the ride comfort of vehicles. *Strojniški vestnik - Journal of Mechanical Engineering*, vol. 53, no. 4, p. 233-240.

[12] Guclu, R. (2004). The fuzzy-logic control of active suspensions without suspension-gap degeneration. *Strojniški vestnik - Journal of Mechanical Engineering*, vol. 50, no. 10, p. 462-468.

[13] Lin, J., Lian, R.J., Huang, C.N., Sie, W.T. (2009). Enhanced fuzzy sliding mode controller for active suspension systems. *Mechatronics*, vol. 19, p. 1178-1190.

[14] Eski, I., Yıldırım, S. (2009). Vibration control of vehicle active suspension system using a new robust neural network control system. *Simulation Modelling Practice and Theory*, vol. 17, p. 778-793.

[15] Lin, J.S., Huang, C.J. (2004). Nonlinear backstepping active suspension design applied to a half-car model. *Vehicle System Dynamics*, vol. 42, no. 6, p. 373-393.

[16] Yagiz, N., Hacioglu, Y. (2008). Backstepping control of a vehicle with active suspensions. *Control Engineering Practice*, vol. 16, p. 1457-1467.

[17] Stojic, M.R. (1998). *Digital control systems*, 4th ed. Faculty of Electrical Engineering – Belgrade. (in Serbian)

[18] Mathworks (2008). *http://www.mathworks.com,* accessed on 2008-11-29.

[19] Luo, A.C.J., Rajendran, A. (2007). Periodic motions and stability in a semi-active suspension system with MR damping. *Journal of Vibration and Control*, vol. 13, no. 5, p. 687-709.

[20] Lee, D., Allan, J., Thompson, H., Bennett, S. (2001). PID control for a distributed system with a smart actuator. *Control Engineering Practice*, vol. 9, p. 1235-1244.

[21] Cominos, P., Munro, N. (2002). PID controllers: recent tuning methods and design to specification. *IEE Proceedings - Control Theory and Applications,* vol. 149, no. 1, p. 46-53.

[22] Astrom, K.J., Hagglund, T. (1995). *PID controllers: theory, design and tuning*, 2nd ed. Instrument Society of Automation, USA.

[23] Rogers, J., Craig, K. (2005). On-hardware optimization of stepper-motor system dynamics. *Mechatronics*, vol. 15, p. 291-316.

[24] Curovic, D., Vasic, B., Popovic, V., Curovic, N. (2008). Expert planning of production. *Journal of Institute for Research and Design in Commerce & Industry*, vol. 20, p. 49-58. (in Serbian)