

**EED654E NONLINEAR SYSTEMS**

**Dr. Öğr. Üyesi Mohammed ALKRUNZ**

**Istanbul Aydin University**

**Engineering Faculty, Electrical and Electronics Department**

Hüseyin KURT

**Adaptive Suspension Control Systems**

Adaptive Suspension Control is divided into main two categories. These are active control and passive control. For Passive control, there is no any feedback control. Only damper and sprung are used for damping the road disturbance. But, in active control, there is control input to manage the changing real time. There are many method for active control systems. In literature, the methods are fuzzy logic, PID control, model predictive control, adaptive and robust. The main purpose of suspension control, is to keep the vehicle vertical acceleration at zero or minimize it.

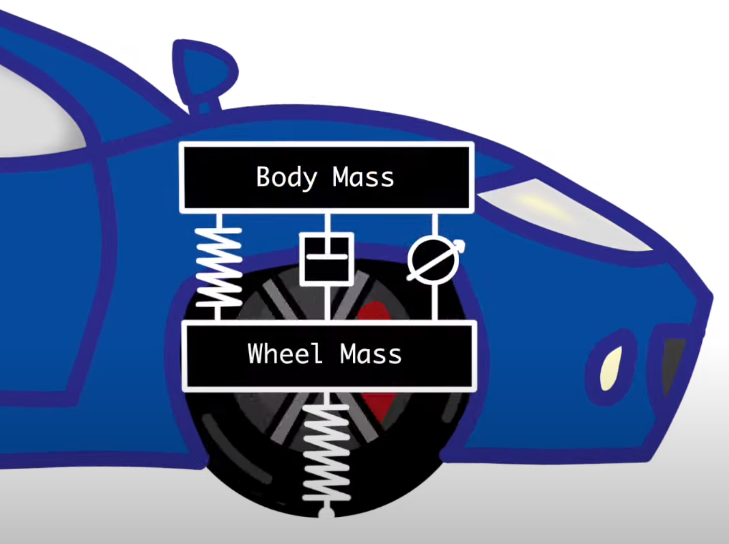


Figure 1: Plant model.

Figure 1 shows the plant model for vehicle suspension model. In this model, there are vehicle sprung which is between vehicle and wheel, wheel sprung which is between wheel and road, damper and control input unit. The control input unit control the distance between vehicle body and wheel. In this study, The mathematical equations are prepared for this plant. The plant model design in Matlab Simulink according to these equations. By using these setup, the plant model was simulated.

1. Mathematical Equations

Equations according to Newton’s second law.

2. Design The Plant Model in Matlab

sim = 10;

ts = 0.01;

t = 0:ts:sim;

ud = zeros(size(t));

ud(1:101) = 0.3\*(1-cos(2\*pi\*t(1:101)));

uc = zeros(size(t));

u = [uc; ud];

This part is for arranging simulation time, number of sample and input variables. Simulation time is 10 second. Sampling frequency is 100. Number of sample is 1000. “ud” stands for road disturbance input. Time of it is 1 second. “uc” is control input. For open loop, it consists of zeros.

m\_car = 300; % kg

m\_wheel = 60; % kg

k\_car = 16000 ; % N/m

k\_wheel = 190000; % N/m

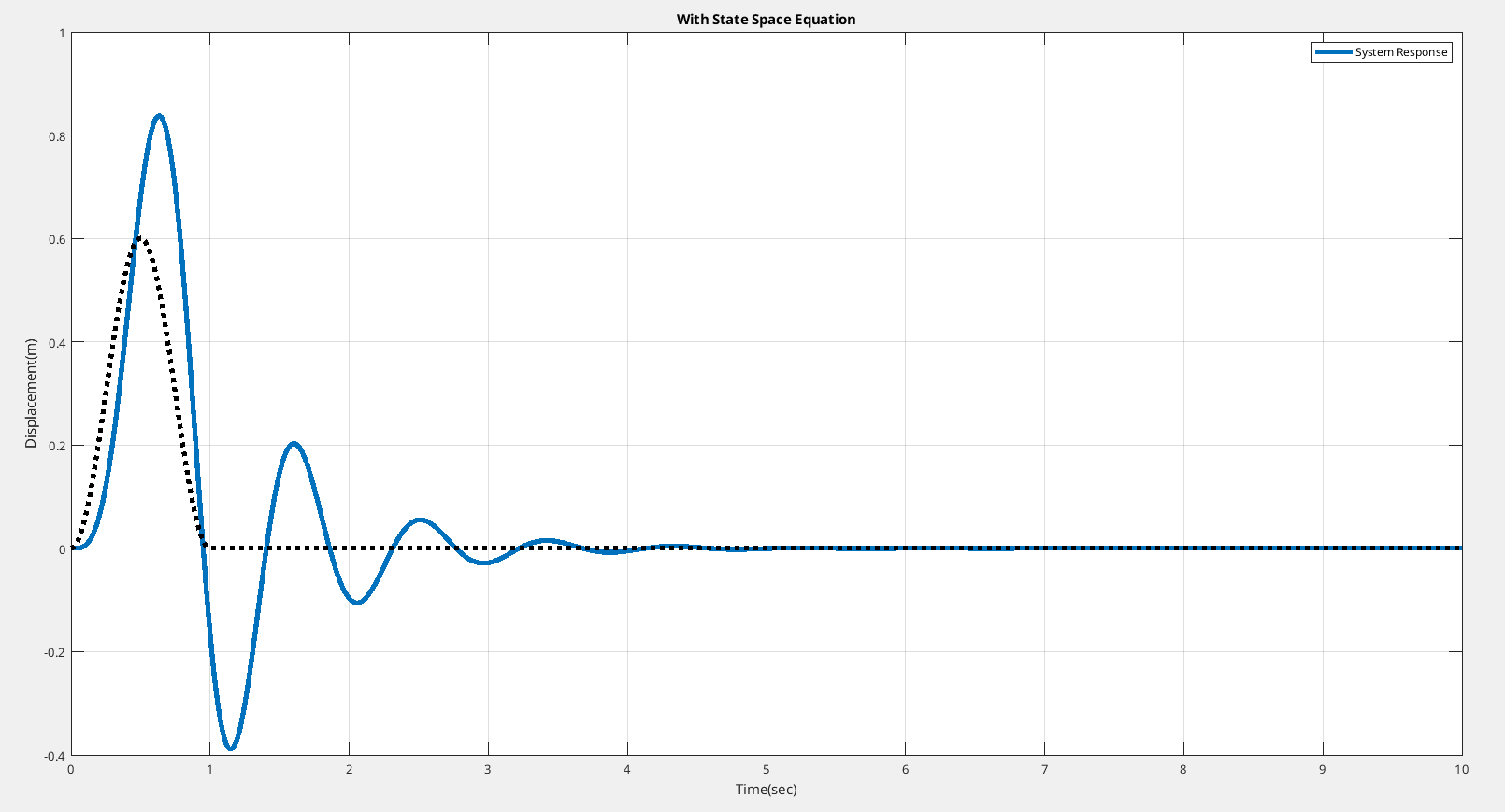
c\_car = 1000; % N/m/s

A = [ 0 1 0 0; [-k\_car -c\_car k\_car c\_car]/m\_car ; 0 0 0 1; [k\_car c\_car (-k\_car-k\_wheel) -c\_car]/m\_wheel];

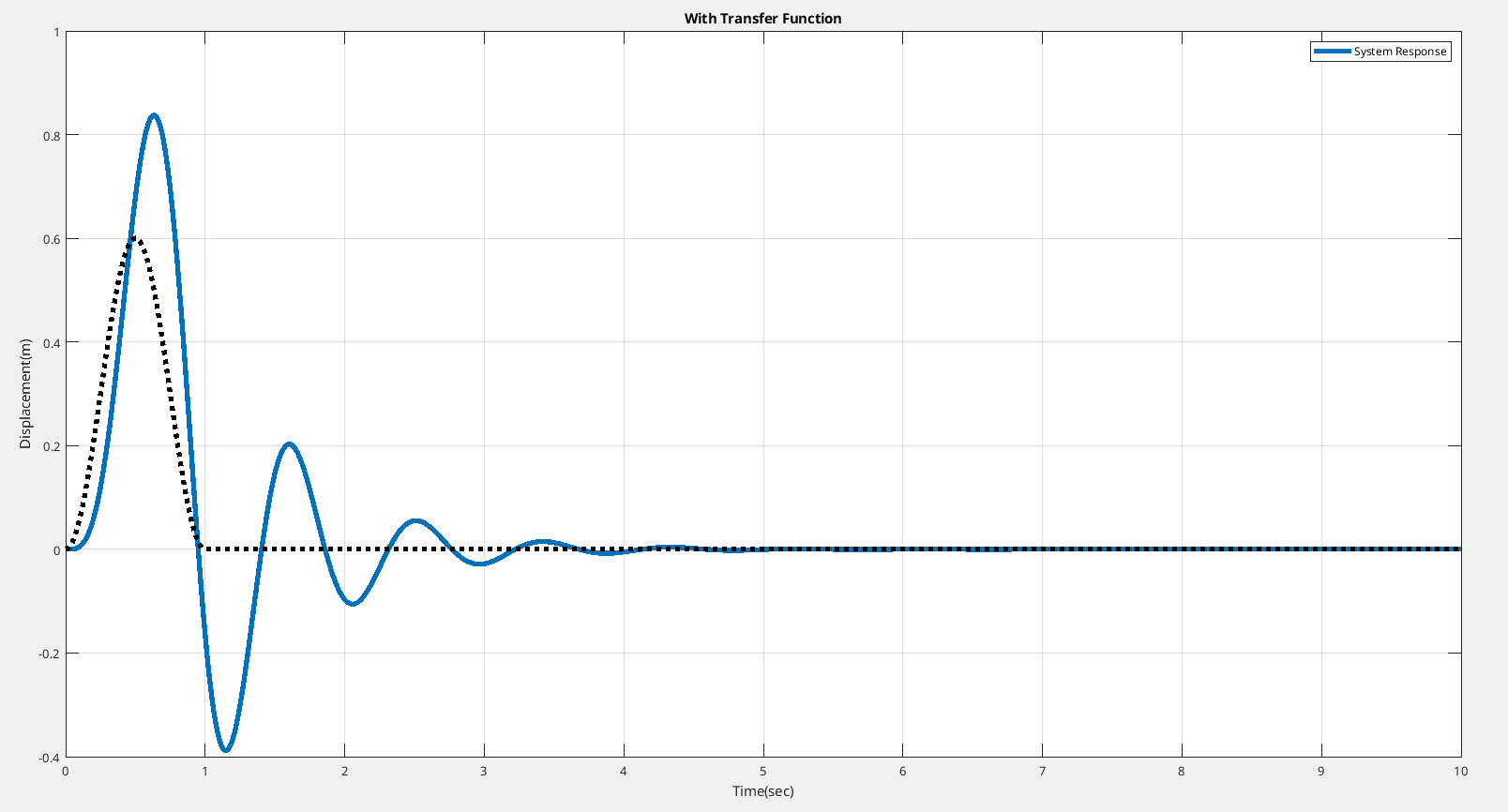
B = [ 0 0; 0 1/m\_car ; 0 0 ; [-1 k\_wheel]/m\_wheel];

C = [1 0 0 0];

D = [0 0];



[b, a] = ss2tf(A,B,C,D,2);



Simulink

