YTU Mechanical Engineering Department

Machine Theory, System Dynamics and Control Division

PID Control Design of a 2DOF Quarter Car Model

Laboratory: Machine Theory, System Dynamics and Control Laboratory, A Block

Laboratory Name: Machine Theory 2

Konu: PID Control Design of a 2DOF Quarter Car Model

Used Devices and Materials

- Computer
- MATLAB-SIMULINK

Theoretical Information:

Reason of the modelling of the physical system is to determine factors of the effects on system Dynamics. Due to the investigated case is machine Dynamics, differancial equations are frequently occured. This approximation creates advantages on investigation of the dynamic system. Analysis Create a vision on the Machine Dynamics. Newton’s law of motion has been used for investigation. These theorem can be applied with reduction of the complex system. With this method, even complex system can be modelled and analyzed.

Suspension System

Vibration analysis is essential approximation to create increment on passenger comfort. Vibration of the vehicle is related from road irregularities to suspension movement. Due to the prediction on vehicle vibration, physical model of the vehicle had to be done. Then mathematical model has been found with Newton’s law. Lastly Simulation has been done and results are occured.

Technological advantage also improve suspension. Active and semi-active suspension applications frequently studied and published. Between passive, semi-active and active suspension system, most reliable displacement control has been occurred with active suspension system. In active suspension system, sensors and force actuators has exist. In according to signal data from suspension (displacement measurement) actuators create opposite force for dissipation. Active control system has been regulated with process with controller unit. With Figure 1. active controlled suspension system has been illustrated.

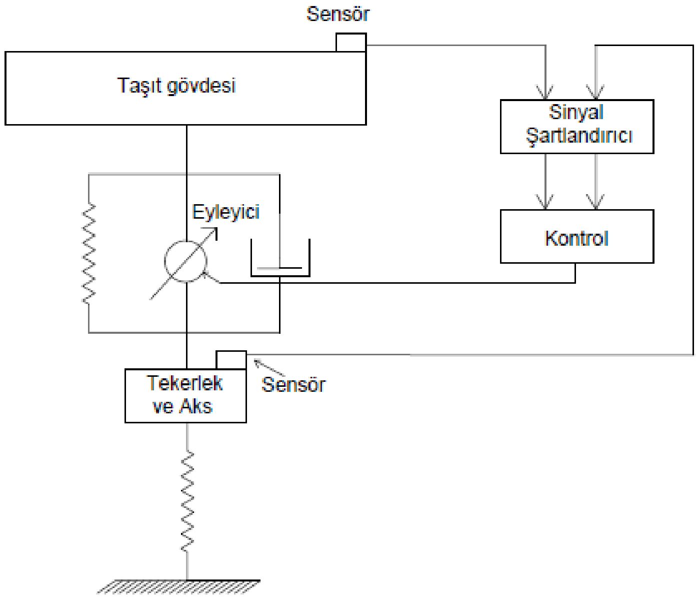


Figure 1. Active Controlled Suspension System Physical Model

Mathematical Model of Quarter Car Model.

Vehicle models can be modelled in according to investigate area. Even complex model can be built with vertical, horizontal and angular movement, simple model are roughly enough, when analyzed subject is narrow. For instanse, for investigating vertical diplacement of the vehicle, 2 DOF quarter car model create meaningful results.

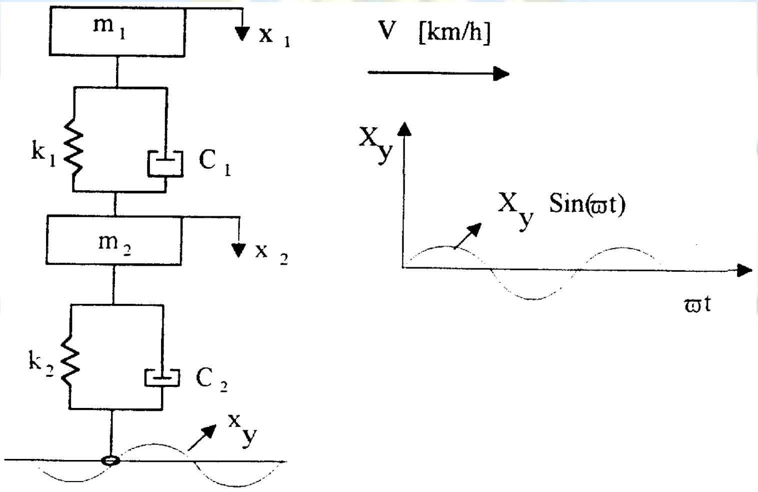


Figure 2. Quarter Car Model

Euler-Lagrange method can be used to obtain motion equation of the quarter car model. Euler-Lagrange method With using energy equation in according to system generalized coordinates, motion equation can be found. Generalized equation defines system’s every each independent movement. All generalized coordinated gives the degree of freedom (DOF) of the system. For detailed information, it can be looked books related the subject. (Francis,1978; Pasin,1984; Rap, 1986)

Euler-Lagrange Equation can be written as

$$\frac{d}{dt}\left(\frac{\partial E_k}{\partial \dot{x}_j}\right)-\frac{\partial E_k}{\partial x_j}+\frac{\partial E_p}{\partial x_j}+\frac{\partial E_D}{\partial \dot{x}_j}=Q_j \qquad (1)$$

Parameters are;

E_k	: Kinetic Energy Equation of the System
E_p	: Potantial Energy Equation of the System
E_D	: Dissipation Energy Equation of the System
Q_j	: Generalized Forces
x_j	: Generalized Coordinates

In model, parameters are:

m_1	: Vehicle's ¼ mass
m_2	: Total mass of Wheel and axle
k_1	: Spring coefficient of suspension
k_2	: Spring coefficient of wheel
c_1	: Damping coefficient of suspension
c_2	: Damping coefficient of wheel
x_1,x_2	: Generalized coordinates of the vehicle
x_y	: Road Irregularities

Kinetic energy equation of the vehicle is:

$$E_k=\frac{1}{2}\left[m_1\dot{x}_1^2+m_2\dot{x}_2^2\right] \qquad (2)$$

Potantial energy equation of the vehicle is:

$$E_p=\frac{1}{2}k_1(x_1-x_2)^2+\frac{1}{2}k_2(x_2-x_y)^2 \qquad (3)$$

Dissipation energy equation of the vehicle is:

$$E_D=\frac{1}{2}C_1(\dot{x}_1-\dot{x}_2)^2+\frac{1}{2}C_2(\dot{x}_2-\dot{x}_y)^2$$

$$m_1\ddot{x}_1+C_1(\dot{x}_1-\dot{x}_2)+k_1(x_1-x_2)=0 \qquad (5)$$

$$m_2\ddot{x}_2+C_1(\dot{x}_2-\dot{x}_1)+k_1(x_2-x_1)+k_2(x_2-x_y)+C_2(\dot{x}_2-\dot{x}_y)=0 \qquad (6)$$

When Euler-Lagrange Equation has been applied on energy equation, (5-6) equations has been calculated. With (7) matrix formation of the motion equation has been given.

$$\begin{bmatrix} m_1 & 0 \\ 0 & \mathrm{m}_2 \end{bmatrix}\begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} + \begin{bmatrix} C_1 & -\mathrm{C}_1 \\ -C_1 & \mathrm{C}_1 + C_2 \end{bmatrix}\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} + \begin{bmatrix} k_1 & -k_1 \\ -k_1 & \mathrm{k}_1 + k_2 \end{bmatrix}\begin{bmatrix} \mathrm{x}_1 \\ \mathrm{x}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ k_2 \end{bmatrix}(x_y) + \begin{bmatrix} 0 \\ C_2 \end{bmatrix}\begin{bmatrix} \dot{x}_y \end{bmatrix}(7)$$

Temel Kontrol İşlemleri

Main reason of the controller is to create a suitable system condition for obtain desired results. In other words, controller changes the system differential equation and its answers. In engineering discipline, there are two types of control system has existed: Open loop controller and close loop controller.

If controller does not create signal in according to system output, then this type of system has been named open loop control system. (Figure 3)

If controller create a signal in according to output of the system, this type of system has been named as closed loop control. With figure 4, reference tracking problem has been illustrated.

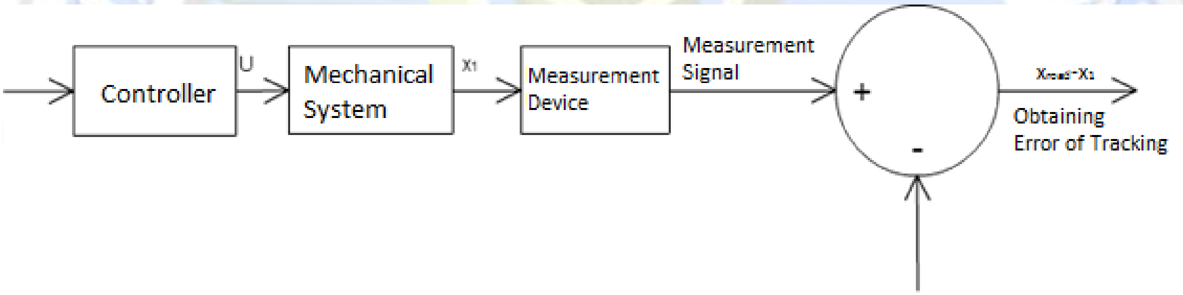


Figure 3. Open Loop Control

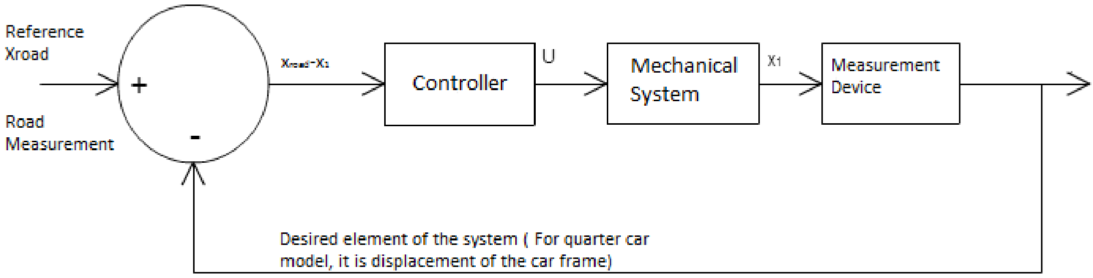

Reference Xroad
Road Measurement
+
-
Xroad-X1
Controller
U
Mechanical System
X1
Measurement Device
Desired element of the system (For quarter car model, it is displacement of the car frame)

Figure 4. Closed Loop Control

System: System is a group, which occurred a result of desired condition with functions and their relationship each other.

Input: Input is a drive energy from outside of the system to the system.

Output: Outputs are the answers of the system

Comparitor: For tracking problem, it has made a comparison between input and output.

Steady State Error: $y_{SS}(t)$

For time domain answers of the system, it gives the final values of the outputs.

$$\lim_{t\to\infty} y_{SS} \neq 0$$

Transient Response: $y_T(t)$

For time domain answers, transient response comes from internal force dynamics. Due to the equiblibrium, it goes to 0 when time is infinite.

$$\lim_{t\to\infty} y_T = 0$$

ve $y(t) = y_T(t) + y_{SS}(t)$

Characteristics of Transient Response of the System

Transient response has been seen all system because of the system's internal convervative energy equiblirium. When a output has been applied, system gain energy, this energy disturbs the equiblirium of the internal energy. Due to the his disturbation, natural behaviour of the system has demonstrated itself. This phenomenon can be called transient response.

This answer is important, because it gives the information about system's behaviour. If this behaviour is fully understood, control of the system become possible. In that cause, step input is widely applied on system to understand this phenomenon and controlled system responses.

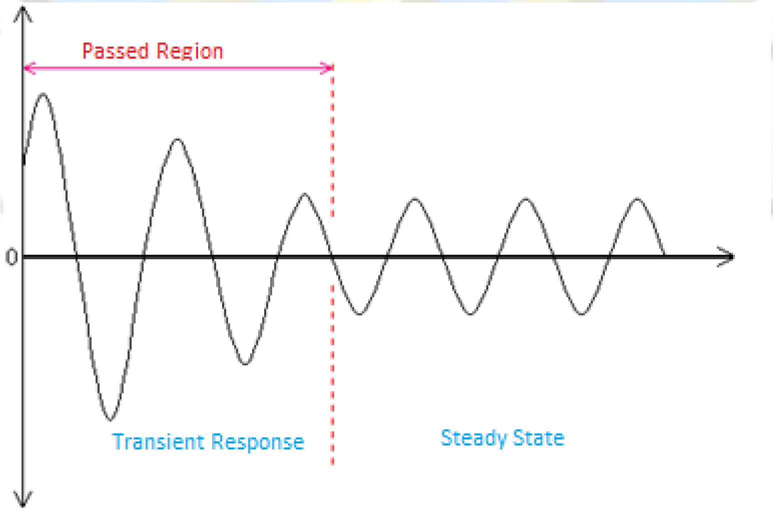


Figure 5. Geçici rejim cevabı karakteristikleri

Delay Time: (t_d) It is the time until begining of the steady state is occured.

Maximum Overshoot Time: (t_p) It is time, when maximum overshoot has seen.

Increasing Time: (t_r) It is the time between %10 and %90 of the maximum overshoot time (t_p)

Maximum overshoot Percent : (M_p) It is a percentance between maximum overshoot and steady state.

$$M_p = \frac{x(t_p) - x(\infty)}{x(\infty)} 100 \;\; [\%]$$

Settling Time: (t_s) It is the time of the begining of the steady state. (Usually, this value has chosen when overshoot percent decreases %2 or %5)

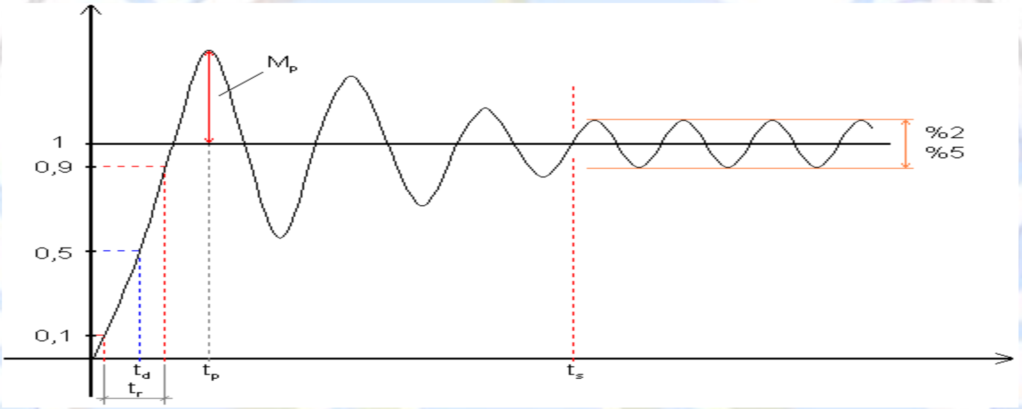


Figure 6. Step Input Answers

Basic Control Elements:

In this section, cascade circuit and its elements has been studied.

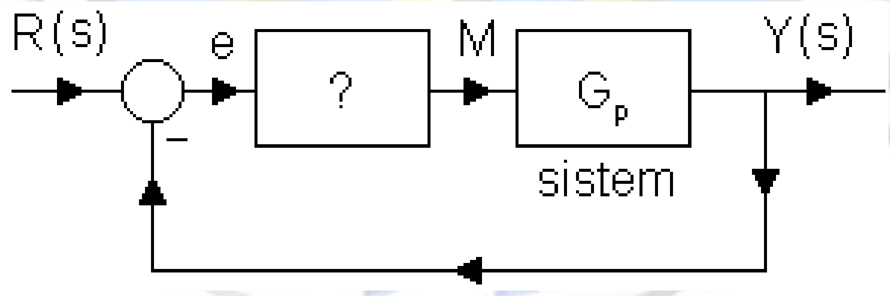


Figure 7. Kontrol sistemi

Proportional Control (P):

In this controller, Error has been multiplied with K_p coefficient, which put on '?'. This controller create a proportion on system chacarteristic equation. Since it is named as 'P'. (K_p: proportional control input)

$M(s) = K_P.E(s) \qquad m(t) = K_P.e(t)$

Characteristics of Proportional Control is,

1) if K_P inceases, sensitivity on parameter change decreases.

2) if K_P inceases, K_P relative stability decreases.

3) if K_P inceases, K_P , response speed increases.

Integral Coefficient (I):

K_I : Integral Coefficient

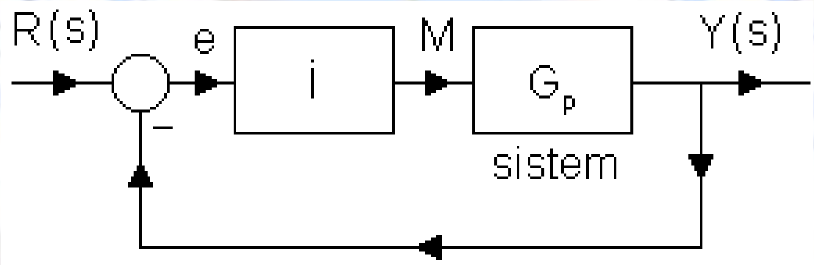


Figure 8. Integral kontrol

$$M(s) = K_I \frac{1}{s} E(s) = \frac{K_I}{s} E(s)$$

$$m(t) = K_I.\int_0^t e(t).dt$$

It controls the steady state error. It defect the effect of future disturbances.

Characteristics of Integral Control

1) It Attenuates steady state error.

2) Stability of the system may be effected.

Derivative Control (D):

K_D : Derivative Control Element $\quad M(s) = K_D.s.E(s) \quad m(t) = K_D.\dot{e}(t)$

NOT: It never used alone. Proportional or Proportional+Integral Control needed.

Characteristics of Derivative Control

1) It attenuates system oscilation.

2) Response speed increase,

3) Settling time decreases.

Proportional-Integral-Derivative Control (PID Control):

It is widely used in industry because of its simplicity and know-how. Conventional Controllers are a time invariant linear feedback controllers. It has not a sensitivity on system parameter change, since adaptation doesn't possible. Mathematical modelling of the system has been obtained and suitable control parameters has been chosen before realization is done. Although, real systems has nonlinear behavior, linearization has been done on the system, with this linearization, conventional controllers can be applied.

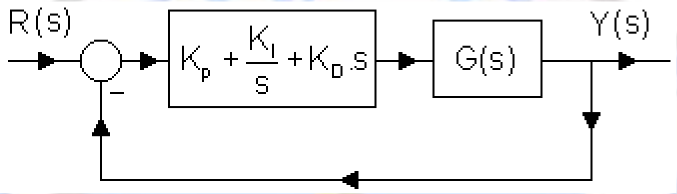


Figure 9. PID Control

For design, several methods has been found to obtain control parameters.

First K_P is found, Then K_D is obtained to create fast response. Lastly, K_I is obtained to attenuate steady state error.

$$M(s) = K_P.E(s) + \frac{K_I}{s}E(s) + K_D.s.E(s)$$

$$m(t) = K_P.e(t) + K_I.\int_0^t e(t).dt + K_D.\dot{e}(t)$$

Conventional PID Controller is well-known method in Industry. Approximately %90 application in industry has been made with this method.

Proportional-Integral-Derivative (PID-Control)

If time response of the PID controller has been investigated,

$$\text{u(t)} = K_P\left[e(t) + \frac{K_I}{K_P}\int_0^1 e(t)dt + \frac{K_D}{K_P}\frac{de}{dt}\right] \qquad (8)$$

is obtained. Then, it can be simplify like,

$$\text{u(t)} = K_P\left[e(t) + \frac{1}{T_i}\int_0^1 e(t)d(t) + T_d\frac{de}{dt}\right] \qquad (9)$$

Lastly, If transfer function of the system has been found, controller can be applied on system. Transfer function of the system is,

$$\frac{U(s)}{E(s)} = K_P\left(1+\frac{1}{T_i s}+T_d s\right) \tag{10}$$

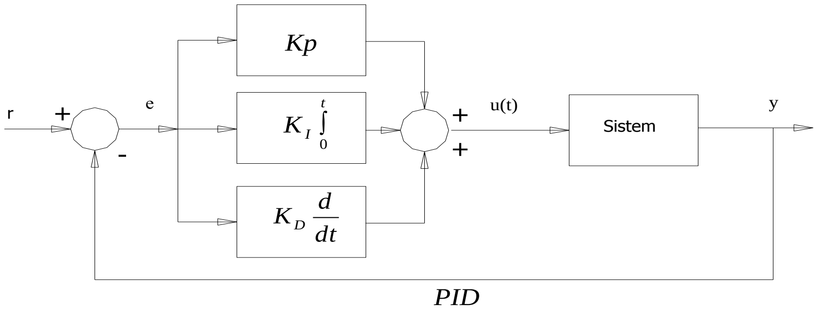


Figure 10. PID Controller

PID control provides three characteristics of three controller. With that, fast, reliable and stabile controller can be designed. PID controller is more complicated than other controllers (PI and PD). In this controller, K_p, T_i and T_j parameters has need to tuned correctly. If it is not provided, this superior characteristics cannot be observed (Kuo, 1995).

Basic Features of PID Controller

In below, transfer function of the PID Controller can be reshown,

$$G_d(s) = \frac{U(s)}{E(s)} = K_P\left(1+\frac{1}{T_i s}+T_d s\right) \tag{11}$$

- P- Control : It has been used, when simple and cheap controller design is desired in industry. Because of this condition, P controller has used only simple, basic tasks.

- PI- Kontrol: Widely used in Pressure, Flow and level control. Integral control attenuates steady state error. Since PI controller has been used approximate %75-90 percent in PID controller users. Also, PI controller design is cheap and simple. It can provide reliable results, when reference tracking problem has been solved with PI-Controller. If P is enough to control, I Controller is not essential.

- Delay problem of the settling can be solved with derivative controller. D controller can be used with P and PI controllers. It is used on servo system, which shows undesiring transient responses like robot applications. Also Derivative controller can be used on unstable or marginally stable system. With this use, system gains damper effect. Major problem of application of derivative controller, it might also increase oscillation because of creating negative damping effect.

Application of PID Controller on 2 DOF Quarter Car Model

Application of PID Controller on 2 DOF Quarter Car Model has been made which is shown Figure 11. Wheel damping ratio has been chosen as 0 to provide simplicity.

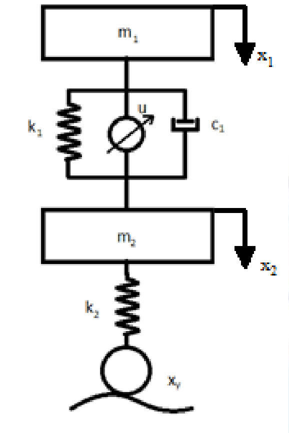


Figure 11. 2 DOF Quarter Car Model

Total Kinetic Energy of the system is,

$$E_k = \frac{1}{2}\left[m_1\dot{x}_1^2 + m_2\dot{x}_2^2\right] \tag{12}$$

Total Potential Energy of the system is,

$$E_p = \frac{1}{2}k_1\left(x_1 - x_2\right)^2 + \frac{1}{2}k_2\left(x_2 - x_y\right)^2 \tag{13}$$

Total Damping Energy of the system is,

$$E_D = \frac{1}{2}C_1\left(\dot{x}_1 - \dot{x}_2\right)^2 \tag{14}$$

With using this functions, application of Euler-Lagrange equation gives the motion equation of the system. U has been defined as a generalized force, which can be defined as an external force.

Equations:

$$m_1\ddot{x}_1 + c_1(x_1 - x_2) + k_1(x_1 - x_2) = -U \tag{15}$$

$$m_2\ddot{x}_2 + c_1(x_2 - x_1) + k_1(x_2 - x_1) + k_2\left(x_2 - x_y\right) = U \tag{16}$$

MATLAB- SIMULINK UYGULAMASI

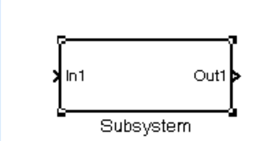
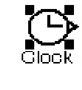
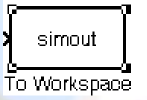
In this application, PID control of a 2DOF quarter car model has been done. Application has made on MATLAB-Simulink platform, which is widely used system identification, analysis and control design. After obtaining mathematical model of the physical model, MATLAB coding has needed. In below, step by step application has been introduced.

1. Open the **“Matlab”**
2. On home tab in MATLAB, find the **“Simulink Library Browser”** and open it.
3. On opening window, find file and choose **“create a new model”** icon and create a Simulink Environment

Save this document in a new folder. Name must be chosen with English characters [Sample: ‘’iki_sd_tasit”]

4. On this Simulink Environment, we will build the simulation. We will find all block what we need, **“Simulink Library Browser”** and put these blocks in Simulink Environment. In table 1, needed blocks list and their appearance has given. With mause’s left click, put them in the saved environment.

Table 1. Simulink Block-First Level

Name	**Simulink Library**	**Block Apperance**
. Subsystem	Ports-Subsytems	
. Clock	Sources	
. To Workspaces	Sinks	

Subsystem: It is used for the simplification of the Simulink appearance. With subsystem block, group of blocks can be shown just one block. It also useful, when some configuration of the subsystem has been reused. If ‘’input’’ or ‘’output’’ block has been put, input for input block and output gate for output block is occurred on subsystem block. With this, communication can be provided between the blocks.

Clock: It creates a ‘’double variable’’. In this double variable, Simulink Environment Time has been written in according to sample time.

To Workspace: It creates a defined variable and transfers it From Simulink to MATLAB Workspace. With It, Simulink arrays can be written in MATLAB can be plot or reoriented in MATLAB Editors. This block create a ‘’struct’’ format, when Default setup is used. It can be very useful when making multiple task, however, in this experiment, single tasks has been used. In that cause, setup of the block must be change to ‘’double’’ format.

5. After putting the block, connect **“Clock”** and **“To Workspace”** from the gates of the block with using left click of the mouse. Then open the **“Subsystem”** environment with

double click. Then Delete the input and output blocks with keyboard ‘’delete’’ button. With this application, Block has be seen like below, which is given Figure 12.

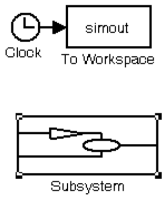
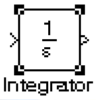
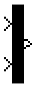
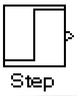
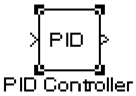
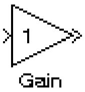
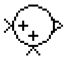


Figure 12. Appearance of the Simulation After 5th step.

6. In table 2, Needed blocks has been given, which is used in Subsystem Environment.

Table 2. Controlled Vehicle Model’s Simulation Blocks

Name	Simulink Library	Block Apperance
. Fnc	User-Defined Functions	
. Integrator	Continuous	
. Mux	Signal Routing	
. To Workspaces	Sinks	simout To Workspace
. Step	Sources	
. PID	Simulink Extras/Additional Linear	
. Gain	Math Operations	
. Sum	Math Operations	

Fcn: It allow to write Mathematical equality. Also it can be used Workspace variables.

Integrator: This blocks take integral of the connected variable.

Mux: It creates an U array with all connected signals. As an example, mux creates 5 input gates, signals are written as u[1], u[2], u[3], u[4], u[5] column vectors. To setup mux input, double click the block, then specify how much input do you need.

Step: It creates step function. Step finish value, step time can be setup.

PID: PID controller block can be defined with P, I and D parameters from the inside of the block. With it, PID controller can be designed.

Gain: Multiplying operation

Sum: Add to substract operation can be done. To setup this, open the block and specify the operation symbol like (++, or +-) Every Symbol create another input gate.

7. In this model, **2 "Fnc" block and 4 integrator blcok** has needed. Because, we have two motion equation, also, this motion equation can be functionalize with acceleration. It can be shown as $\ddot{x} = \ddot{x}(x_1, \dot{x}_1, x_2, \dot{x}_2, x_y, U)$

   Blocks can be recreate, when mouse right click hold them and release. To apply this method, hold the blocks with mouse right click then release it some empty space in Simulink environment.

8. **"Mux"** is setup with 6 input. Because every function has 6 independent variable. Every input in mux define the function input. Main idea is to define every mux input as acceleration function's independent parameters. This parameters is defined Fcn input for every time interval. As a result, Figure 13 is obtained.

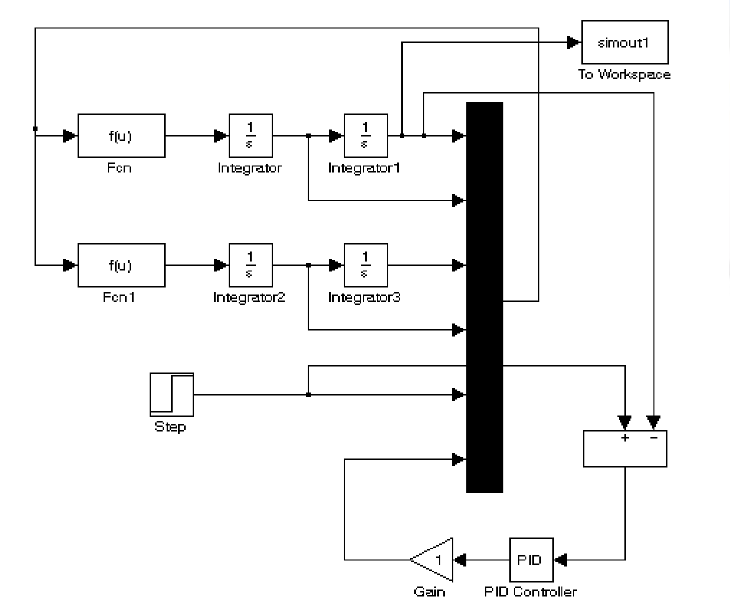


Figure 13 Simulation Configuration of the Mux for Differential Equation Solve

“Gain” and “PID” blocks are flipped with **“Format-Flip Block”** because of the obtain suitable position for the signal flow. **“Sum”** has been convert **“rectangular”** from the inside of the block, then from **“list of signs” “+ -.”** Symbols has been setup for substract. From **“To Workspace”** change the **“save format”** to **“Array”** All “To Workspace” has need to be same configuration.

9. In fcn block, acceleration function has to be written. With integration block, velocity and displacement can be found. This velocity and displacement replace to function’s velocity and displacement. In Figure 14, mux configuration and meaning of the every signal has been given. u[1] is displacement of the m1 mass , u[2] velocity of the m1 mass. Similarly, u[1] is displacement of the m1 mass , u[2] velocity of the m1 mass. u[5] is the road irregularities and u[6] is the controller signal.

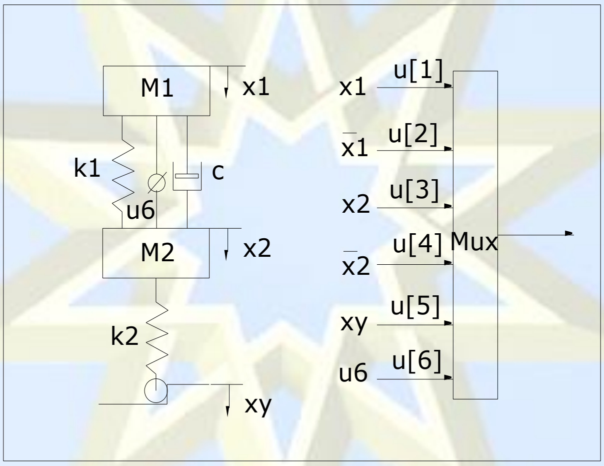


Figure 14 Mux array format

If motion equations rearrange as an acceleration function, it convert to this form:

$$\ddot{x}_1 = (-\frac{1}{m_1})(c_1(x_1 - x_2) + k_1(x_1 - x_2) + U) \tag{17}$$

$$\ddot{x}_1 = (-\frac{1}{m_2})(c_1(x_2 - x_1) + k_1(x_2 - x_1) + k_2\left(x_2 - x_y\right) - U) \tag{18}$$

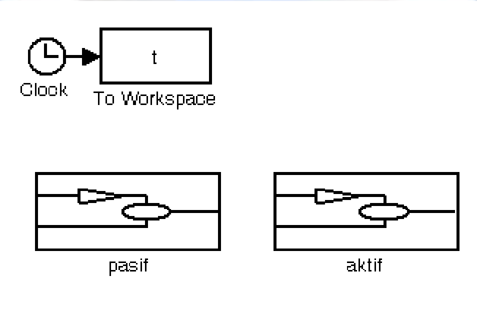
With MATLAB Syntax, it can be written like below.

Fcn) (-1/m1)*(c1*(u[2]-u[4])+k1*(u[1]-u[3])+u[6]) (19)

Fcn1) (-1/m2)*(c1*(u[4]-u[2])+k1*(u[3]-u[1])+k2*(u[3]-u[5])-u[6]) (20)

10. In **“Step”** block, **“Final Value”** has setup as **0.05** With this configuration, vehicle road irregularity goes to 0.05 step.

11. Open the **PID**. Write P, I and D to Proportional (P), Integral (I) ve Derivative (D), which can be seen inside of the PID Block.
12. Inside of the "Subsytem", open **"To Workspace"** and change the **"variable name"** (Example: **x1**). Similarly , Clock's connected **"To Workspace"** variable name has to be changed **"t"**. with that, displacement of the m1 mass has defines as "x1" variable and time has been defined with "t" variable.
13. In first level of the Simulink environment, subsystem has been dublicated with using right click hold and release. Then Change their name as **"pasif"** and **"actif"**. Figure 15 demonstrated their appearance. In pasif subsystem, gain of the PID control has change to 0.



Şekil 15 Appearance of Pasif and Aktif Simulation

14. **Open the "aktif"** subsystem and change the **"To Workspace" "variable name from "x1" to "x1c"**
15. From MATLAB interface, find the editor tab and open a new MATLAB script. In MATLAB Script, lets write this code.

```
clc
clear all
%% Bu kısım Taşıtın Parametrelerini Vermek için kullanılmıştır.
m1=290; % AnaGövde Kütlesi
m2=40; % Süspansiyon Kütlesi
c1=1000; %Süspasiyon Sönümü Katsayısı
k1=13000; %Süspansiyon Yay Katsayısı
k2=220000; %Tekerlek Yay katsayısı
P=13000; % Orantı (Proportional) Parametresi
I=0.1; % Integral (Integral) Parametresi
D=0.001;  % Türev (Derivative) Parametresi
%% Bu kısım Simülasyonu çözer ve  Grafikleri çizer.
sim('iki_sd_tasit') %Simülasyonu Çalıştırır
figure(1) %Figure 1'i açar
plot(t,x1,'black',t,x1c) %grafik çizme x1 datasını t ye bağlı siyah çizer. x1c'yi ise mavi çizer.
legend('Yer Degisimi','Kontrolcülü'); %Bilgilendirme Notlarını açar
xlabel('zaman (s)') %Apsis Bilgisi
ylabel('konum (m)'); %Ordinant Bilgisi
```

16. This m-file has to be saved in same folder, which 'iki_sd_tasit' Simulink file is located. Then press the **“debug-run”** button and start it. In MATLAB script, parameters has been obtained from workspace, then Simulink environment has started with “sim” command. It also can be possible in Simulink Window. Because of first configuration of P=1,I=1 and D=0, coincident plots has been observed. In Figure 16, x1 displacement can be seen.

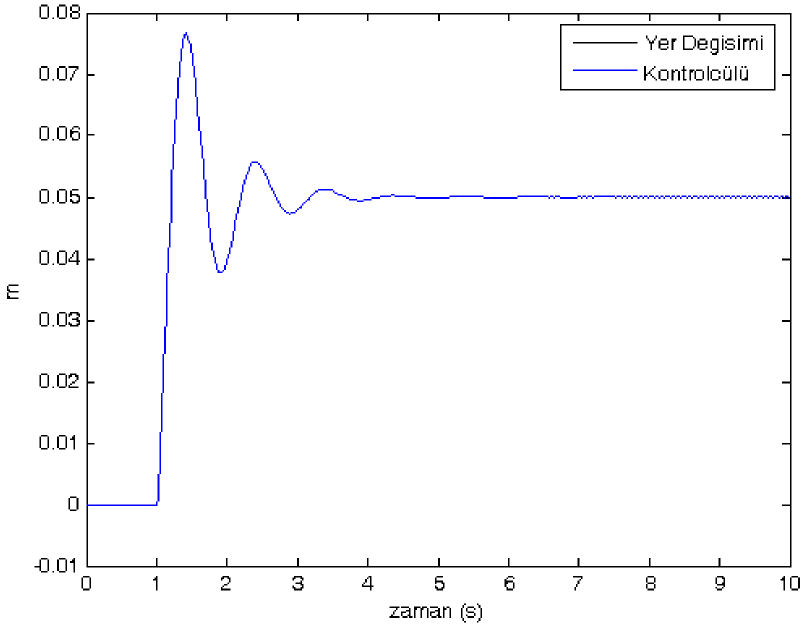


Figure 16 P=1 I=1 D=0 , displacement of m1 mass (x1)

17. In .m file, let’s change the P,I and D parameter as **P=1000, I=1, D=1**. With this configuration, .m file restarted. In Figure 17, x1 displacement can be seen.

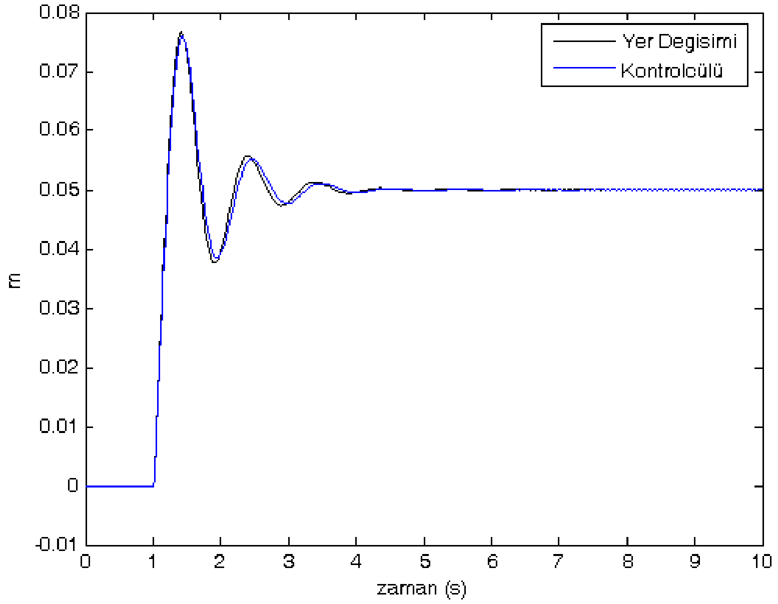
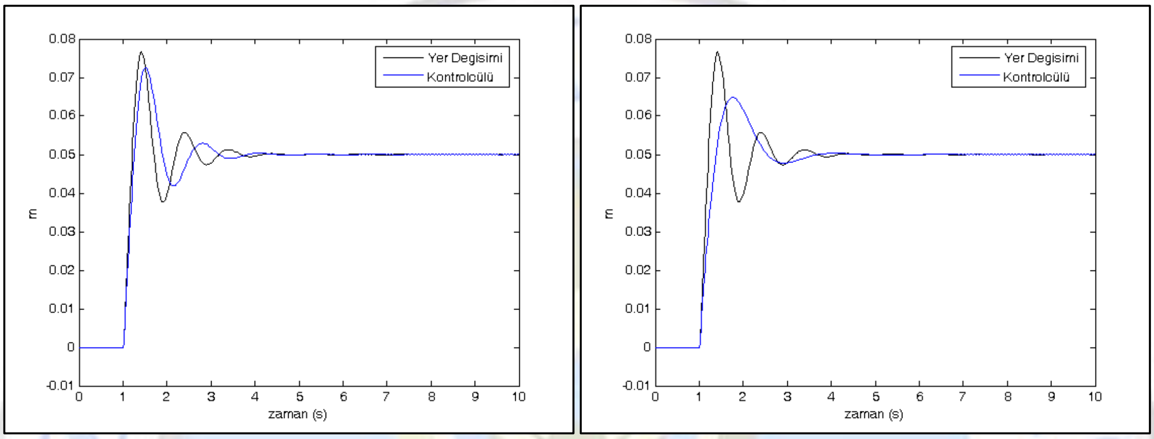


Figure 17 P=1000 I=1 D=0 , displacement of m1 mass (x1)

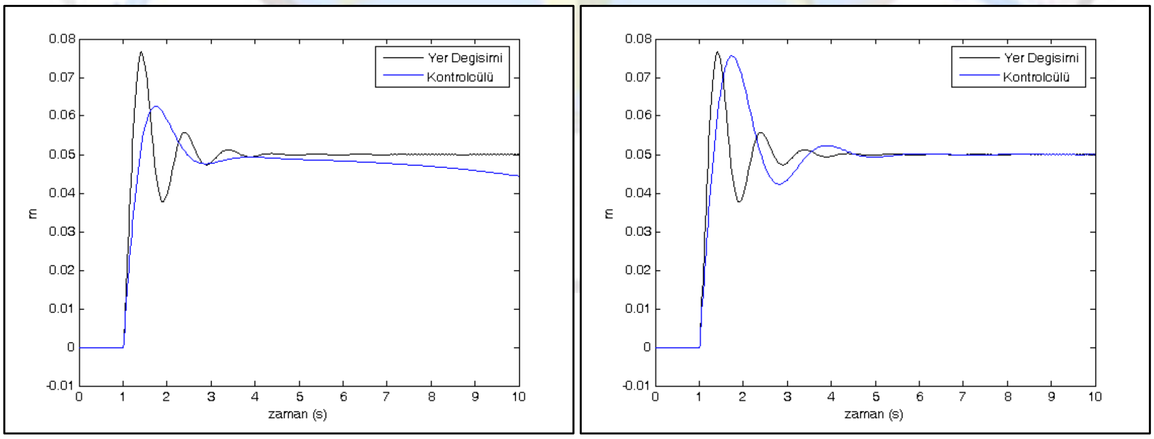
18. In .m file, let’s change the P,I and D parameter as **P=5000, I=1, D=1**. With this configuration, .m file restarted. In Figure 18a, x1 displacement can be seen. Then also, In .m file, let’s change the P,I and D parameter as **P=10000, I=1, D=1**. With this configuration, .m file restarted. In Figure 18b, x1 displacement can be seen.



A B

Figure 18 a) P=5000 I=1 D=0 , displacement of m1 mass (x1) b) P=10000 I=1 D=0 , displacement of m1 mass (x1)

19. Let’s change the parameter as P=10000, I=1000, D=1 and restart the simulation. Figure 19.a gives the result of the simulation. Then, lets change the parameter as P=10000, I=1, D=300. Figure 19.b gives the result of the simulation.



A B

Figure 19 a) P=10000 I=1000 D=0 , displacement of m1 mass (x1) b) P=10000 I=1 D=300 , displacement of m1 mass (x1)

20. Lastly with the parameters of P=13000, I=0.1, D=0.001 best result can be obtained. As you can see, smooth and fast result can be achieved with right PID Controller parameters. Figure 20 gives the result of the this simulation.

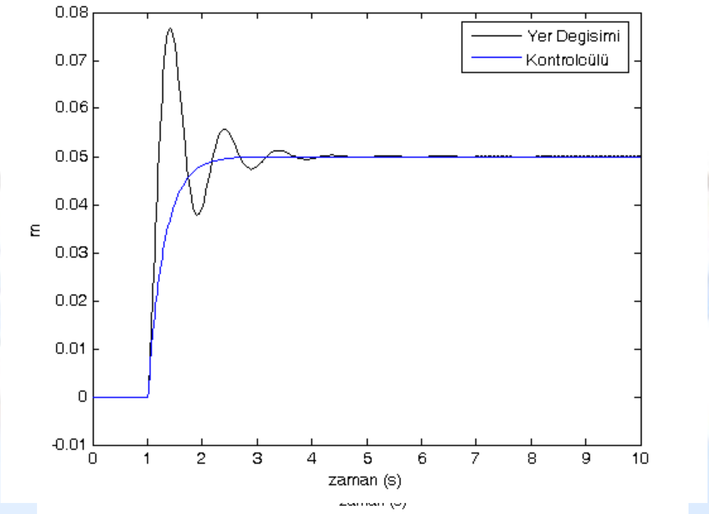


Figure 20 P=13000 I=0.1 D=0.001 , displacement of m1 mass (x1)