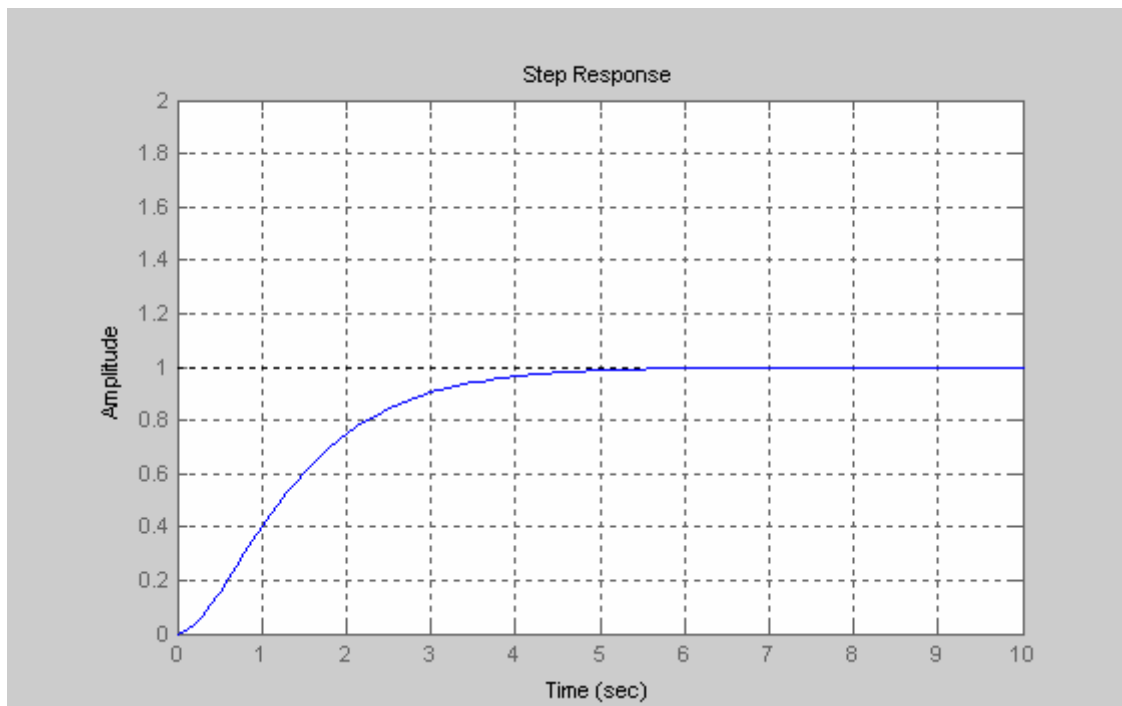


컴퓨터 예제 4-2

이 예제의 단위 계단 응답은 MATLAB을 이용하여 다음과 같이 그릴 수 있다.

```
>> g=tf(2,[1 3 2]);  
>> step(g,10)  
>> axis([0 10 0 2])  
>> grid on
```

단위 계단 응답을 그려주는 step 함수에서 최종 시간을 지정할 수 있다. 이 경우에는 10초로 지정하였다. 그림을 그린 후, axis 명령에 의해서 가로 축과 세로 축의 범위를 지정할 수 있다. 그린 결과는 다음과 같다.



컴퓨터 예제 4-3

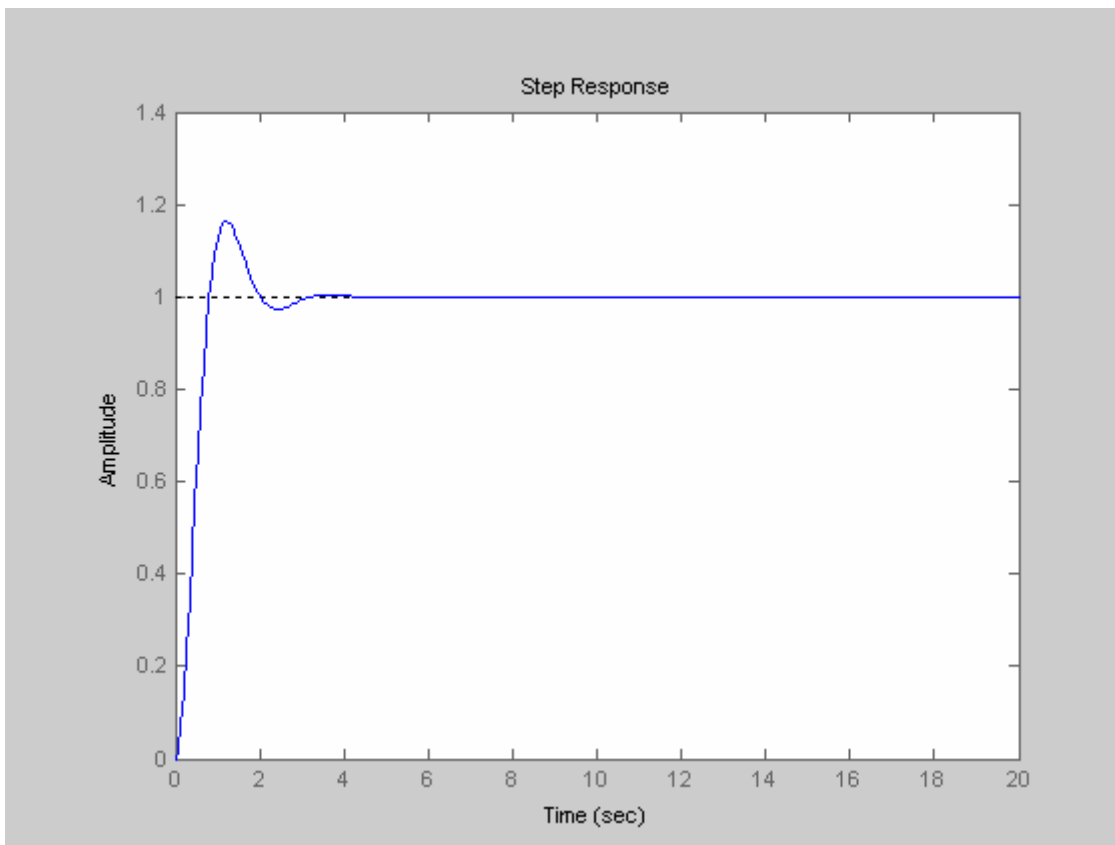
이 예제에서는 페루프 시스템의 단위 계단 응답을 구한다. 전체 페루프 전달 함수는 다음과 같이 feedback 함수를 이용하여 구할 수 있다.

```
>> M=2;
```

```

>> B=6;
>> K=18;
>> G=tf(1,[M B 0])
Transfer function:
    1
-----
 2 s^2 + 6 s
>> GT=feedback(K*G,1)
Transfer function:
    18
-----
 2 s^2 + 6 s + 18
>> step(GT,20)

```



컴퓨터 예제 4-4

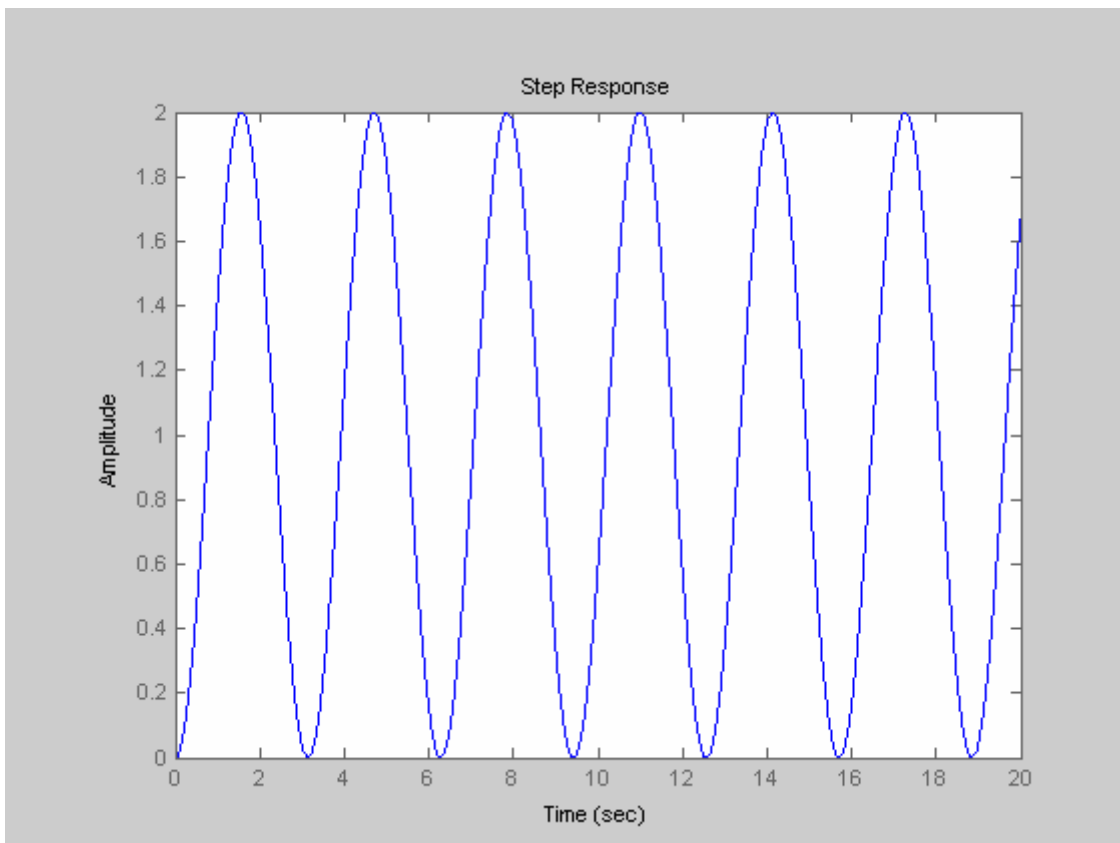
이 예제의 응답을 그리기 위한 명령은 다음과 같다.

```

>> M=0.25;
>> K=1;
>> G=tf(1,[M 0 K])
Transfer function:
      1
-----
0.25 s^2 + 1
>> t=0:0.001:20;
>> step(G,t)

```

단위 계단 응답을 그리는 함수 step에서, 최종 시간 대신에 시간 벡터를 지정하여 그릴 수 있다. 위의 예에서는 시간 벡터를 0초에서 20초까지 0.001초 간격으로 만들어서 단위 계단 응답을 그렸다. 이와 같이 하면 그림 그리는 간격을 조밀하게 하여 그래프를 좀더 부드럽게 그릴 수 있다.



질량의 값을 바꾸어 그리면 다음과 같다.

```

>> M=1;

```

```
>> G=tf(1,[M 0 K])
```

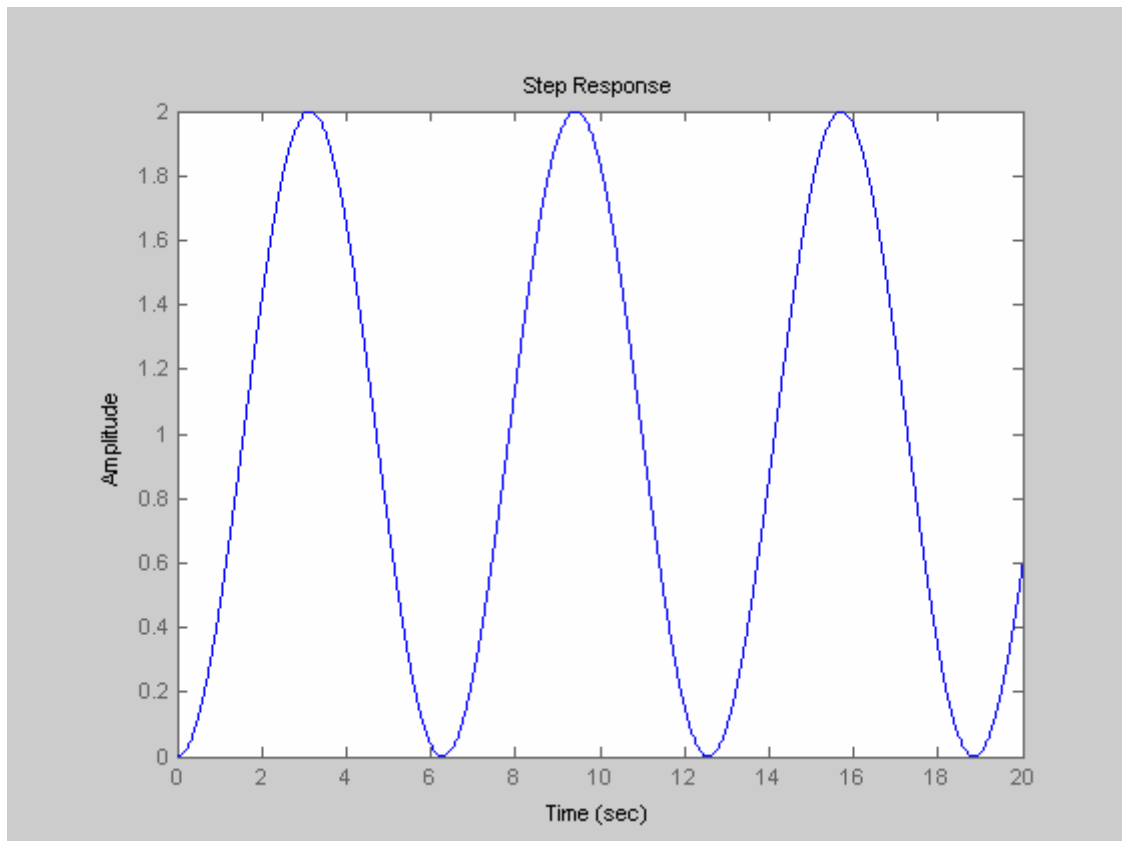
```
Transfer function:
```

```
1
```

```
-----
```

```
s^2 + 1
```

```
>> step(G,t)
```



컴퓨터 예제 4-5

예제 4-3과 같은 방법으로 그리면 다음과 같다.

```
>> B=-0.6;
```

```
>> K=18;
```

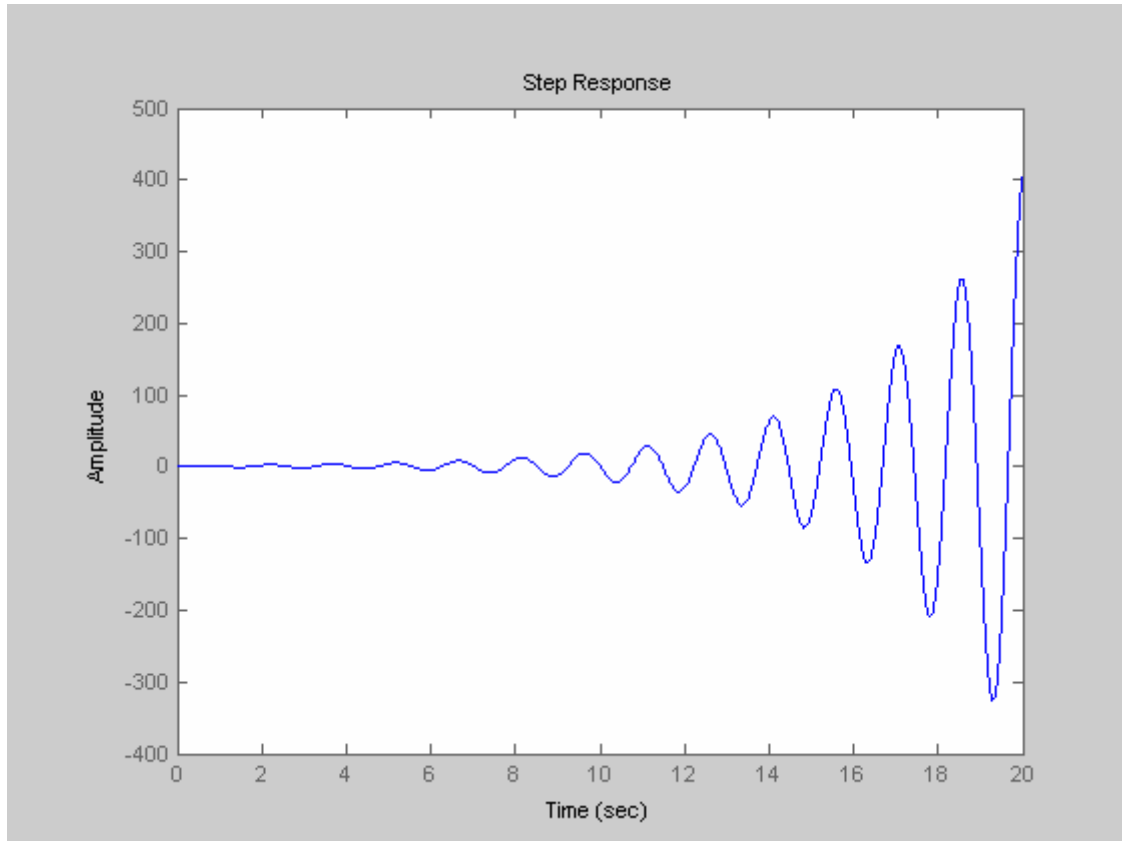
```
>> G=tf(1,[M B 0]);
```

```
>> GT=feedback(K*G,1)
```

```
Transfer function:
```

```
18
```

```
-----  
s^2 - 0.6 s + 18  
>> step(GT,t)
```



컴퓨터 예제 4-6

이 예제의 방정식의 근을 구하면 다음과 같다.

```
>> p=[1 2 3 4 5 2];  
>> roots(p)  
ans =  
    0.3571 + 1.3262i  
    0.3571 - 1.3262i  
   -1.0627 + 0.8194i  
   -1.0627 - 0.8194i  
   -0.5888
```

컴퓨터 예제 4-7

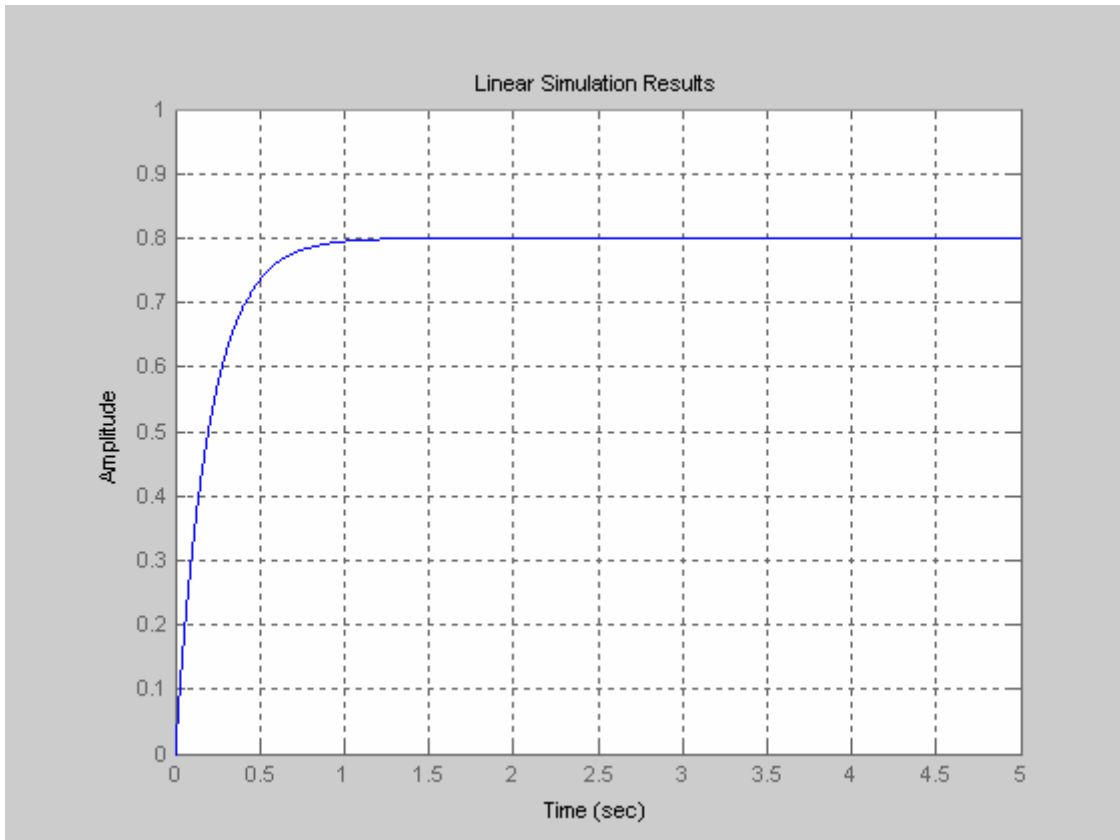
이 예제의 방정식의 근을 구하면 다음과 같다.

```
>> p=[1 1 3 2 4 2 2];
>> roots(p)
ans =
    0.4551 + 1.0987i
    0.4551 - 1.0987i
   -0.4551 + 1.0987i
   -0.4551 - 1.0987i
   -0.5000 + 0.8660i
   -0.5000 - 0.8660i
```

컴퓨터 예제 4-10

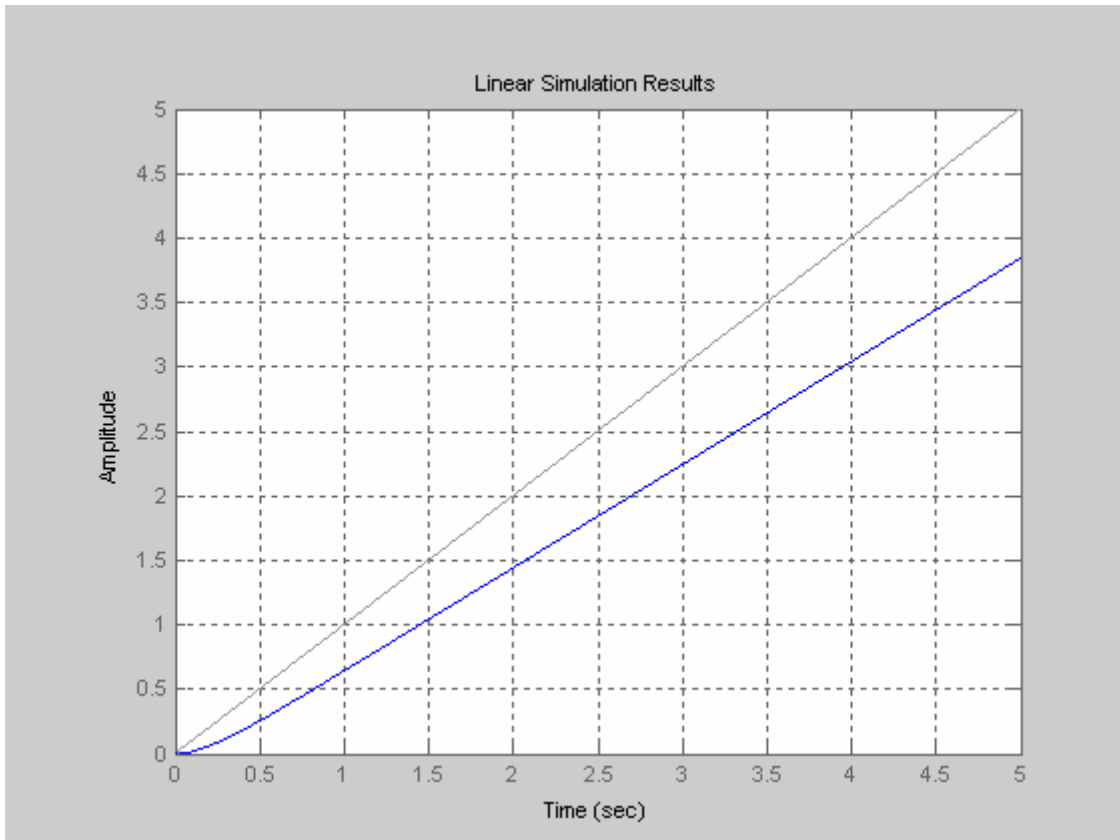
이 예제에서는 step 대신에 lsim 함수를 이용하여 시뮬레이션을 실행해 본다. 단위 계단 명령 입력에 대한 시뮬레이션은 step 함수를 이용하면 편리하지만, 단위 계단이 아닌 형태의 명령 입력에 대한 시뮬레이션은 lsim 함수를 이용한다. 아래에서는 시간 변수 벡터를 만든 후, 시간 변수 벡터의 크기와 같은 명령 입력을 R=ones(size(t))와 같이 만들어서 lsim 함수에 입력하여 단위 계단 응답을 그린다.

```
>> K1=2;
>> A=2;
>> tau=1;
>> G=tf(K1*A,[tau 1]);
>> GT=feedback(G,1)
Transfer function:
    4
-----
s + 5
>> t=0:0.001:5;
>> R=ones(size(t));
>> lsim(GT,R,t)
>> grid on
```



단위 경사 명령 입력에 대한 응답은 lsim을 이용하여 다음과 같이 그릴 수 있다.

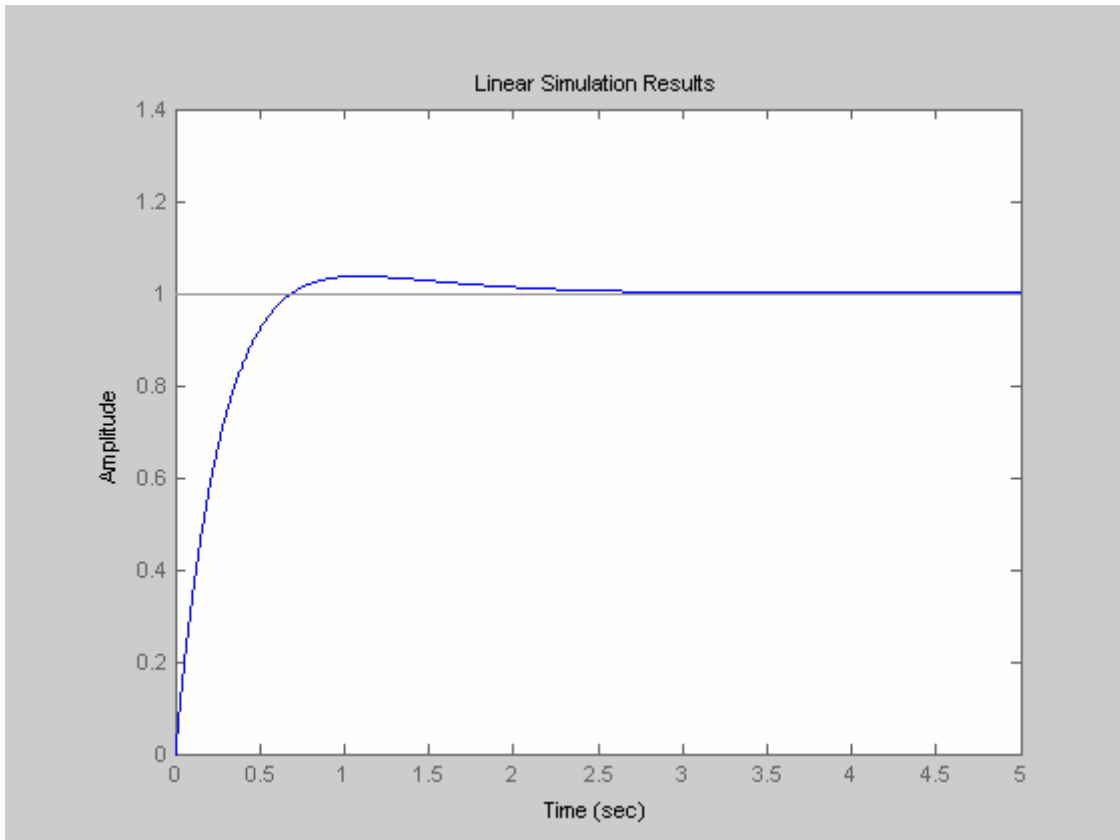
```
>> K1=2;
>> A=2;
>> tau=1;
>> G=tf(K1*A,[tau 1]);
>> GT=feedback(G,1)
Transfer function:
    4
-----
s + 5
>> t=0:0.001:5;
>> R=t;
>> lsim(GT,R,t)
>> grid on
```



컴퓨터 예제 4-11

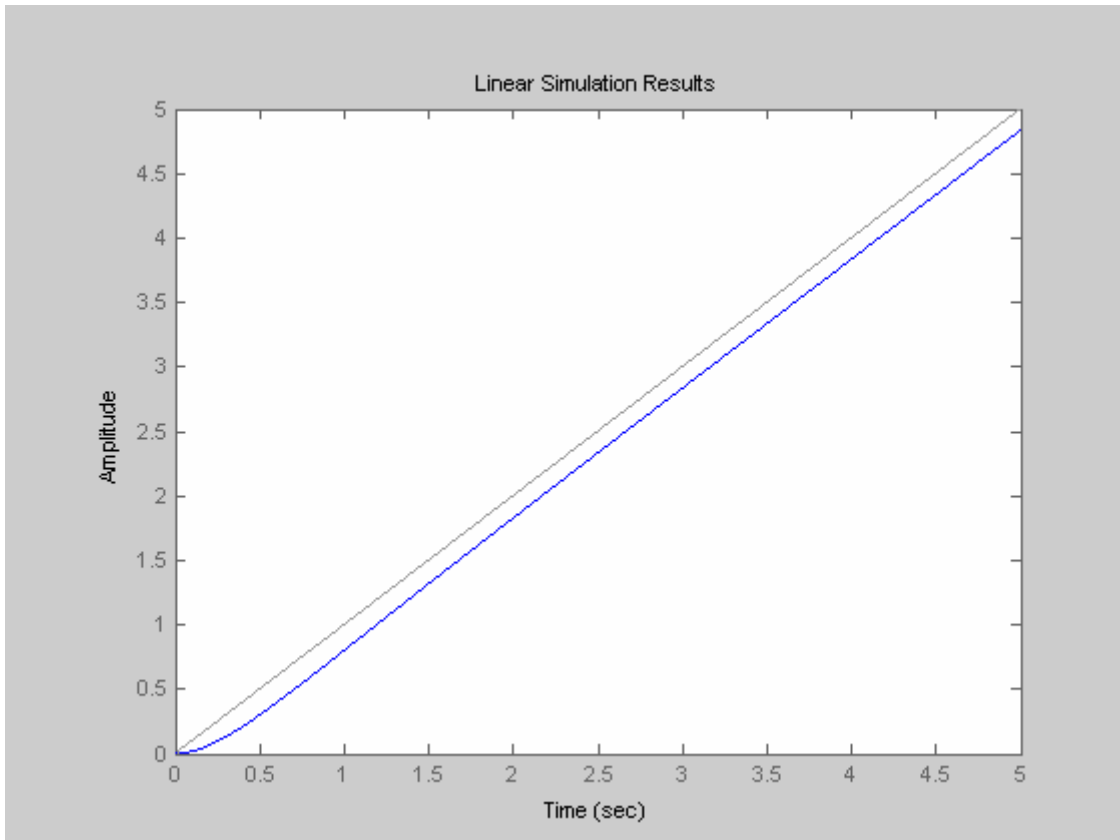
단위 계단 응답은 다음과 같다.

```
>> K1=2;K2=3;A=2;tau=1;
>> G=tf(A*[K1 K2],[tau 1 0]);
>> GT=feedback(G,1)
Transfer function:
    4 s + 6
-----
s^2 + 5 s + 6
>> t=0:0.001:5;
>> R=ones(size(t));
>> lsim(GT,R,t)
```

단위 경사 응답은 다음과 같다.

```
>> K1=2;K2=3;A=2;tau=1;
>> G=tf(A*[K1 K2],[tau 1 0]);
>> GT=feedback(G,1)
Transfer function:
    4 s + 6
-----
s^2 + 5 s + 6
>> t=0:0.001:5;
>> R=t;
>> lsim(GT,R,t)
```



단위 포물선 응답은 다음과 같다. 아래에서 $t.*t$ 는 t 벡터의 각 항을 제공한 것이다.

```
>> K1=2;K2=3;A=2;tau=1;
>> G=tf(A*[K1 K2],[tau 1 0]);
>> GT=feedback(G,1)
Transfer function:
    4 s + 6
-----
s^2 + 5 s + 6
>> t=0:0.001:10;
>> R=t.*t/2;
>> lsim(GT,R,t)
```

Linear Simulation Results

