Question 1:

a) Consider the following objective function
min f(x)=(x–1)^2(x–2)(x–3) , 0 <= x <=4

i) Use the method of Newton-Raphson to compute x* that minimizes f (x) for the following three initial values of x0 = 0.2,1.5 and 2.5 with 10–10 tolerance.(Write matlab m-file code)

ii) Plot the change of the x versus iterarion number , the alternation of the objective function by the evolution of x and gradient information versus iterarion number. (Write matlab m-file code)

Algorithm:

Step 1: Initialize x_0, ε and $k=0$

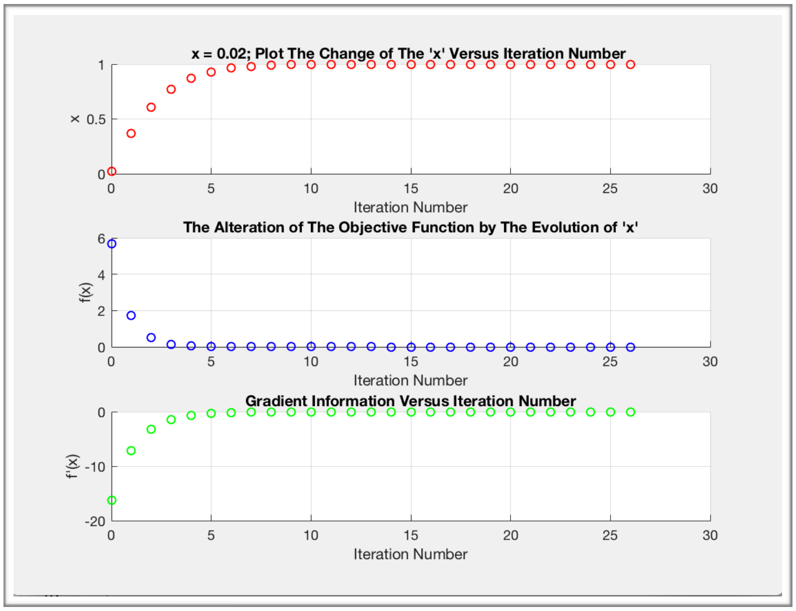
Step 2: Calculate $\Delta x_k = -\dfrac{f'(x)}{f''(x)}$

Step 3: Update $x_{k+1} = x_k + \Delta x_k$

Step 4: If $\left|f'(x_{k+1})\right| < \varepsilon$ terminate iteration, else go to **Step 2.**

Yakup Görür (040130052) HW1 MKM-502 Soft Computing

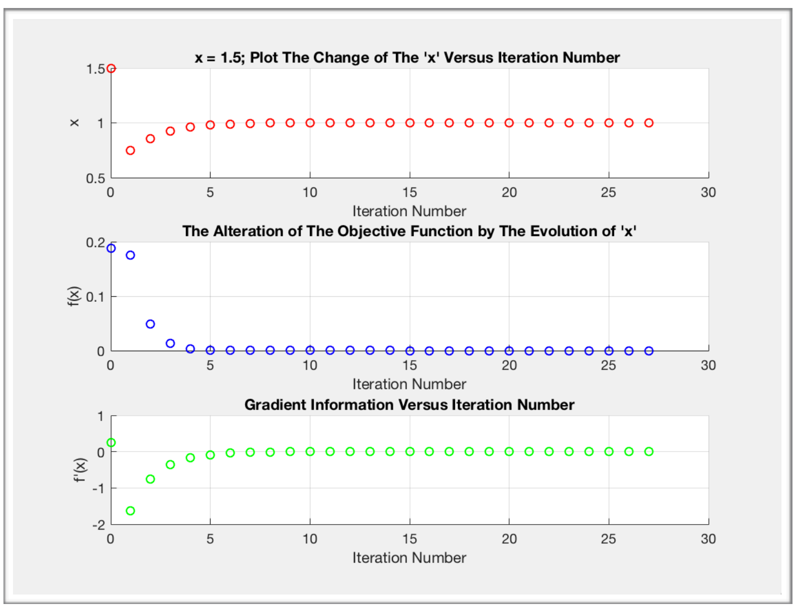
Assoc.Prof.Dr. Gülay ÖKE GÜNEL



x_start= 0.02 Iteration 0: x=0.020, err=0.980
x_start= 0.02 Iteration 1: x=0.367, err=0.632
x_start= 0.02 Iteration 2: x=0.607, err=0.392
x_start= 0.02 Iteration 3: x=0.768, err=0.231
x_start= 0.02 Iteration 4: x=0.869, err=0.130
x_start= 0.02 Iteration 5: x=0.929, err=0.070
x_start= 0.02 Iteration 6: x=0.963, err=0.036
x_start= 0.02 Iteration 7: x=0.981, err=0.018
x_start= 0.02 Iteration 8: x=0.990, err=0.009
x_start= 0.02 Iteration 9: x=0.995, err=0.004
x_start= 0.02 Iteration 10: x=0.997, err=0.002
x_start= 0.02 Iteration 11: x=0.999, err=0.001

x_start= 0.02 Iteration 27:
x=0.99999997617604330369,
err=0.00000002382395669631

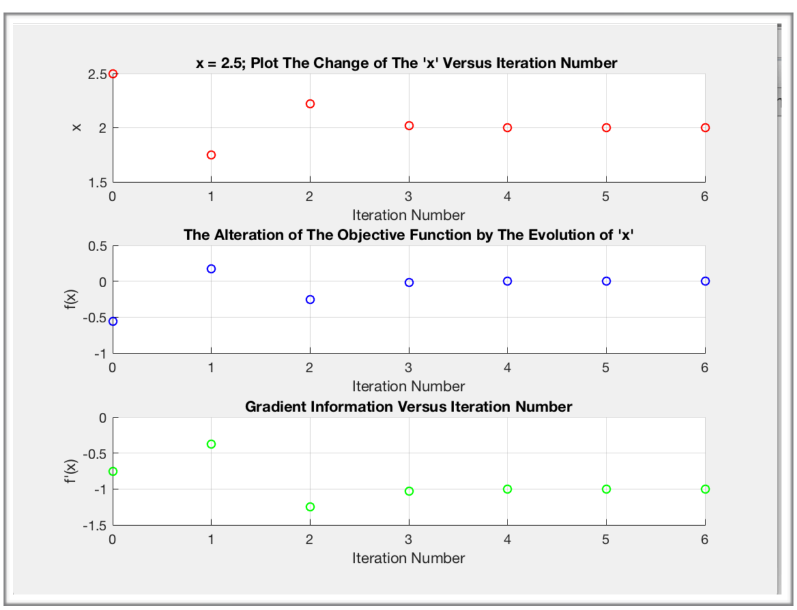
$\epsilon = 10^{-10}$



x_start= 1.5 Iteration 0: x=1.50, err=-0.50
x_start= 1.5 Iteration 1: x=0.750, err=0.250
x_start= 1.5 Iteration 2: x=0.858, err=0.142
x_start= 1.5 Iteration 10: x=0.999, err=0.001
err=0.00000065583238395561

x_start= 1.5 Iteration 28:
x=1.00000001062910159888,
err=-0.00000001062910159888

$\epsilon = 10^{-10}$



x_start= 2.5 Iteration 0: x=2.50, err=-0.50
x_start= 2.5 Iteration 1: x=1.750, err=0.250
x_start= 2.5 Iteration 2: x=2.218, err=-0.2187
x_start= 2.5 Iteration 3: x=2.016, err=-0.016
x_start= 2.5 Iteration 4: x=2.0002, err=-0.00024
x_start= 2.5 Iteration 5: x=2.0001, err=-0.0009

x_start= 2.5 Iteration 6:
x=2.00000000000001332268,
err=-0.00000000000001332268

$\epsilon = 10^{-10}$

b)

Consider the following objective function

$$\min_{x} \; f(x) = (x-1)^2(x-2)(x-3)$$
$$0 \le x \le 4$$

i)Use the method of bisection to compute x^* that minimizes $f(x)$ for $\alpha_a = 1.8$, $\alpha_b = 3$. (Write matlab m-file code)

ii) Plot the change of the x versus iterarion number , the alternation of the objective function by the evolution of x and gradient information versus iterarion number. (Write matlab m-file code)

Algorithm:

Step 1: Determine the interval α_a and α_b when $\alpha_a < \alpha_b$. $\varepsilon = 10^{-4}$

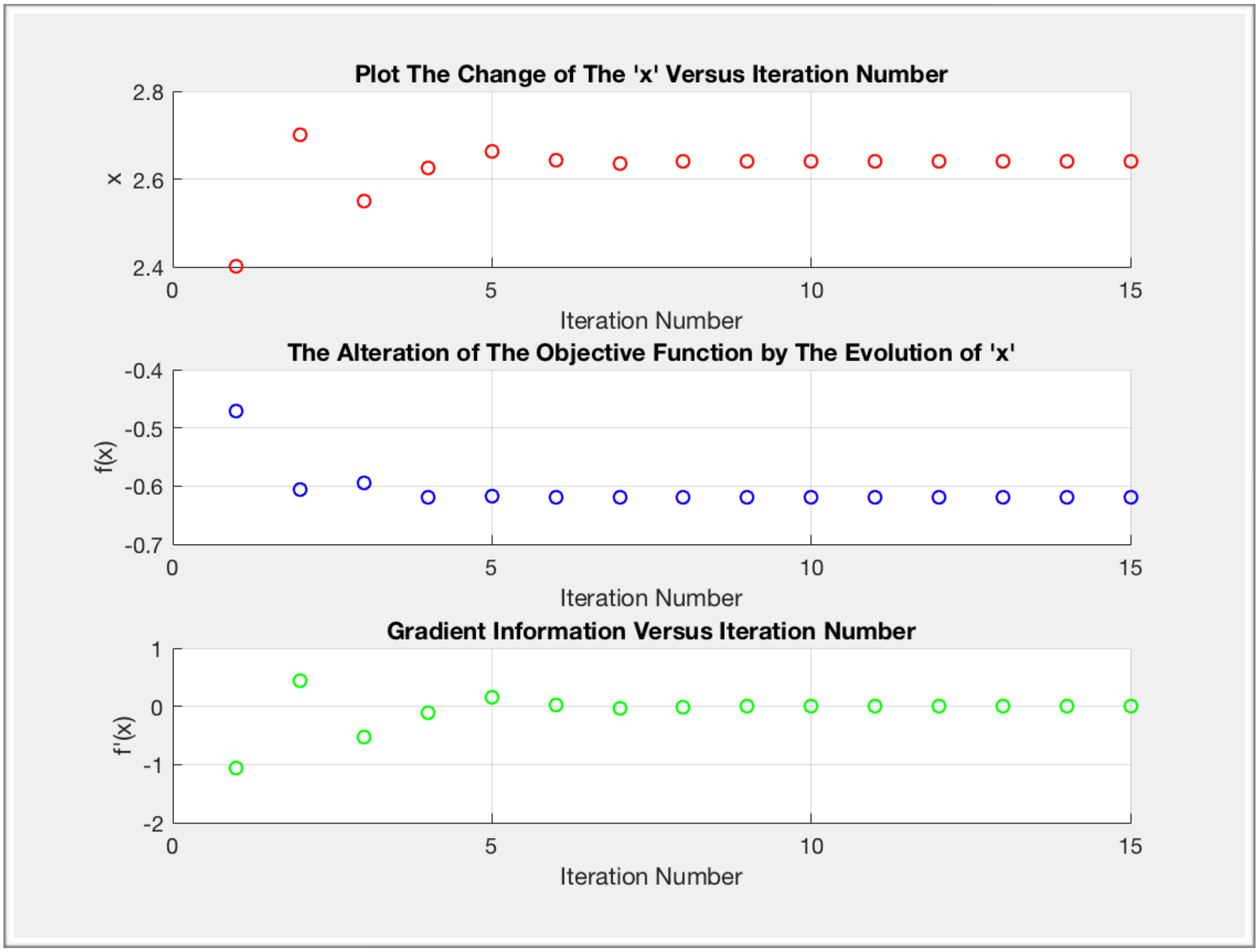
Step 2: Update $\alpha_k = \alpha_a + \dfrac{(\alpha_b - \alpha_a)}{2}$

Step 3: If $f'(\alpha_k) = 0$, terminate the iteration owing to convergence

else and if $(\alpha_b - \alpha_a) < \varepsilon$, then terminate iteration due to tolerans value(ε)

else and if $f'(\alpha_k) f'(\alpha_a) > 0$, then $\alpha_a = \alpha_k$

else $\alpha_b = \alpha_k$ and continue with **Step 2**.



Algorithm on the Matlab

```
iteration_number = 0;
while 1

    %Update a_k (Step 2)
    a_k = a_a + (a_b - a_a)/2;

    %increase the iteration number
    iteration_number = iteration_number + 1;

    %Plot the Figure 1: the change of the x versus iteration number
    plot(ax1, iteration_number, a_k, 'ro');

    %Plot the Figure 2: The Alteration of The Objective Function by The Evolution of x
    plot(ax2, iteration_number, f(a_k), 'bo');

    %Plot the Figure 3: Gradient Information Versus Iteration Number
    plot(ax3, iteration_number, f_derivative(a_k), 'go');

     if f_derivative(a_k) == 0
         fprintf('Terminated iteration owing the convergence\n');
         break;
     elseif (a_b - a_a) < Epsilon
         fprintf('Terminated iteration due to tolerans value(Epsilon)\n');
         break;
     elseif ( f_derivative(a_k) * f_derivative(a_a) ) > 0
         a_a = a_k;
     else
         a_b = a_k;
     end
end
```

Question 2:

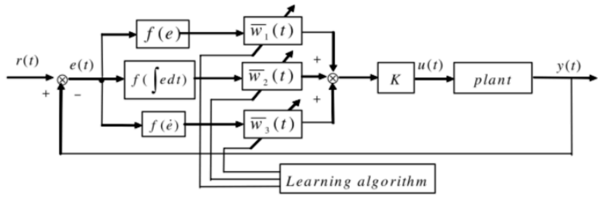


Fig. 1 Neuron based Nonlinear PID Controller

The neuron based nonlinear PID Controller in figure is going to be used to control paper making process. The dynamic characteristics of the plant is as follows:

$$G(z)=\begin{cases}\dfrac{0.2719}{z(z-0.8187)} & , \textit{when } 80\ g/m^2 \textit{ paper is made}\\ \dfrac{0.4484}{z(z-0.7788)} & , \textit{when } 100\ g/m^2 \textit{ paper is made}\\ \dfrac{0.7087}{z(z-0.7165)} & , \textit{when } 120\ g/m^2 \textit{ paper is made}\end{cases}$$

The controller parameters are selected as follows :

$K=1.1,\ \eta_1=15\ ,\ \eta_2=1\ ,\ \eta_3=10$
$\alpha_P=1\ ,\alpha_I=0.5\ ,\ \alpha_D=0.5\ ,$
$\delta_P=0.1\ \ ,\delta_I=0.4\ \ ,\ \delta_D=0.3$

Set the initial values as follows :

$u(0)=0, y(0)=0, w_1(0)=w_2(0)=w_3(0)=0.005$
$e_{tr}(0)=0\ ,\ T_s=0.01\sec(\textit{sampling time})$

a)Plot the response of the system and control signal versus time using the reference signal in figure 2. Plot the alternation of the controller parameters interms of w_1, w_2, w_3 and also K_P, K_I, K_D. (Write matlab m-file code)

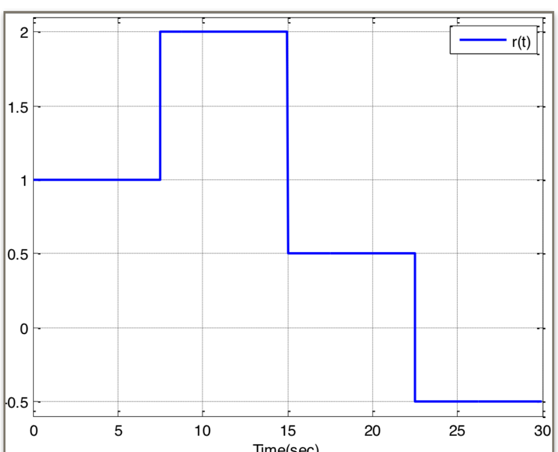


Fig. 2 Reference Signal

Control Algorithm:

Step 0: Set $n=1$

Step 1: Calculate tracking error $e_{tr}(n)=r(n)-y(n)$

Step 2: Calculate inputs of the PID Controller

$$
\begin{aligned}
P(n) &= e_{tr}(n) \\
I(n) &= I(n-1)+e_{tr}(n) \\
D(n) &= e_{tr}(n)-e_{tr}(n-1)
\end{aligned}
$$

Step 3: Calculate outputs of $f(.)$ nonlinear function

$$
\begin{aligned}
x_P(n) &= f(e_{tr}(n),\alpha_P,\delta_P) \\
x_I(n) &= f(\int e_{tr}(n)\,dn,\alpha_I,\delta_I) \\
x_D(n) &= f(\dot{e}_{tr}(n),\alpha_D,\delta_D)
\end{aligned}
$$

Step 4: Calculate control signal

$$
\begin{aligned}
u(n) &= K\,\frac{w_1(n)x_P(n)+w_2(n)x_I(n)+w_3(n)x_D(n)}{w_1(n)+w_2(n)+w_3(n)} \\
&= K_p(n)x_P(n)+K_I(n)x_I(n)+K_D(n)x_D(n)
\end{aligned}
$$

Step 5: Apply control signal to the plant and obtain output of the system

$$y(n+1)=?$$

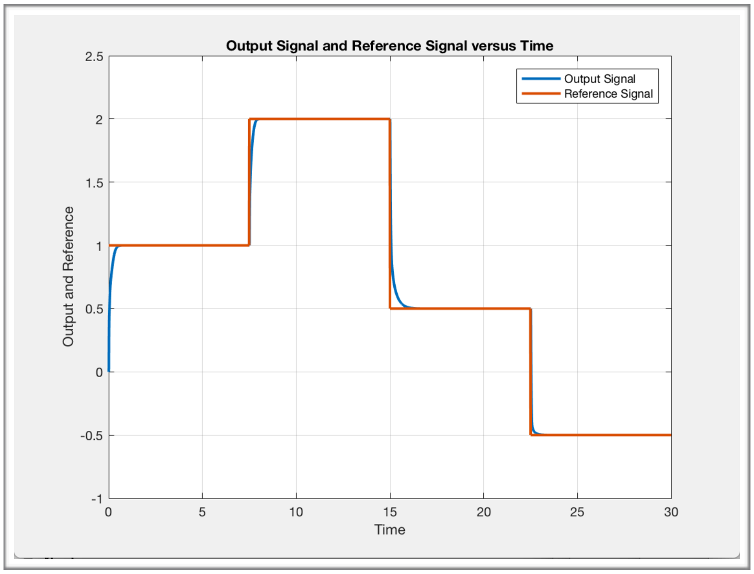
Step 6: Update controller parameters

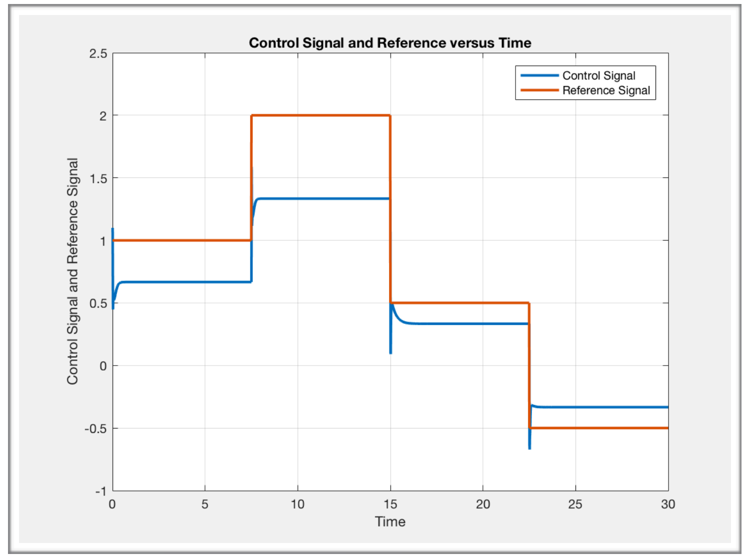
$$
\begin{aligned}
w_1(n+1) &= w_1(n)+d_1e_{tr}(n)u(n)x_P(n) \\
w_2(n+1) &= w_2(n)+d_2e_{tr}(n)u(n)x_I(n) \\
w_3(n+1) &= w_3(n)+d_3e_{tr}(n)u(n)x_D(n)
\end{aligned}
$$

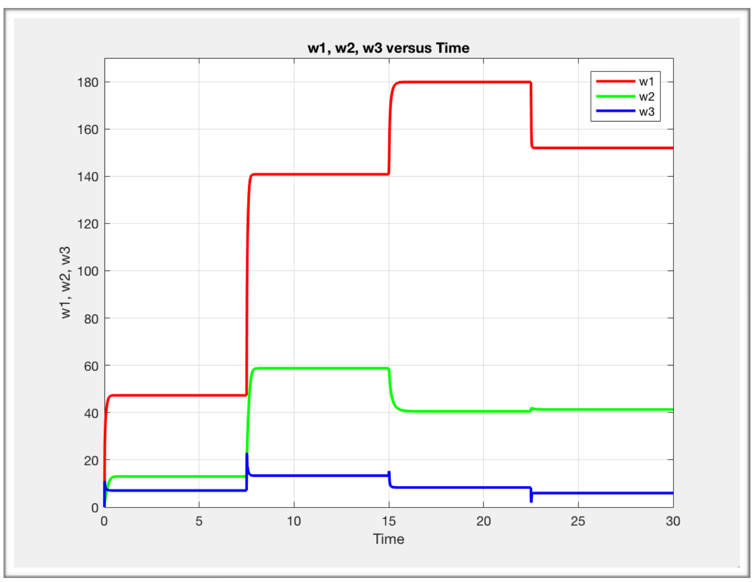
Step 7: $n \leftarrow n+1$ and go to **Step 1.**

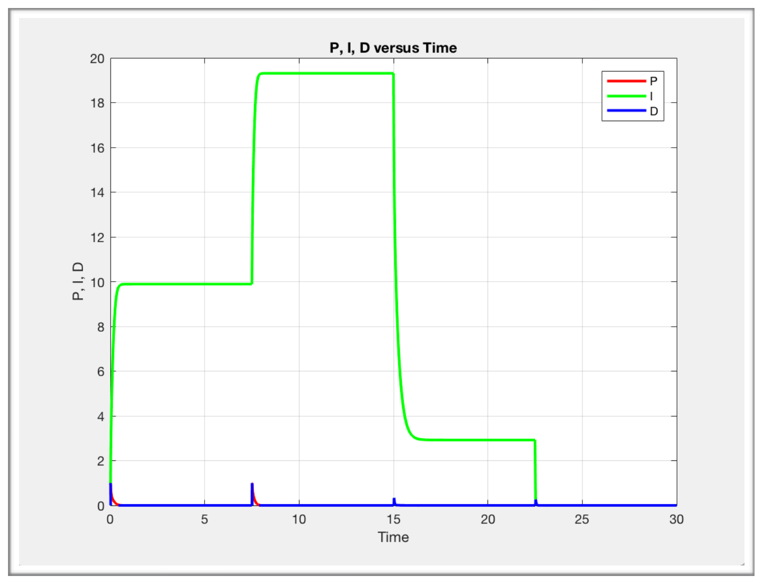
Yakup Görür (040130052) HW1 MKM-502 Soft Computing

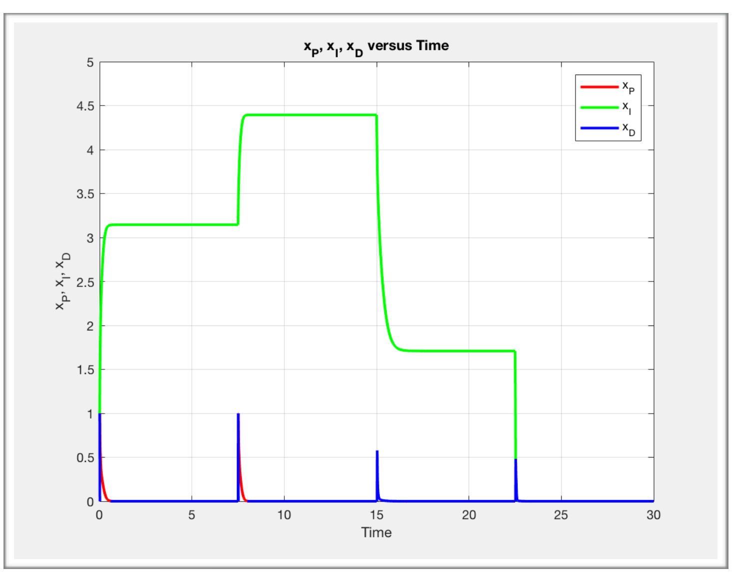
Assoc.Prof.Dr. Gülay ÖKE GÜNEL











b)Plot the response of the system and control signal versus time using the reference signal in figure 3. Plot the alternation of the controller parameters interms of w_1, w_2, w_3 and also K_P, K_I, K_D. ($r(t) = 1 + e^{-\frac{2\pi}{100}t}\sin(t)$) (Write matlab m-file code)

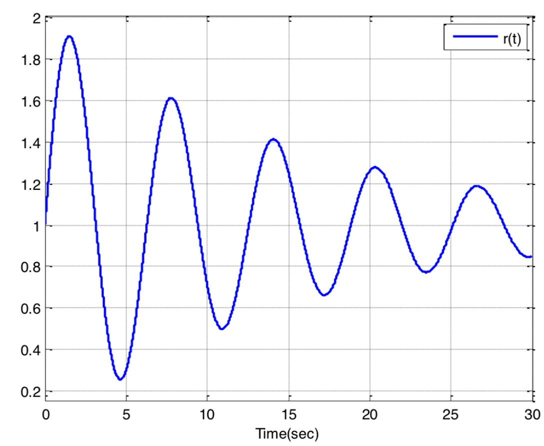


Fig. 3 Reference Signal

Algorithm on the Matlab Code

```
for n=1:3000

    %Step 1: Calculate tracking error etr (n) = r(n) y(n)
    e_tr(n) = r(n) - y(n);

    %Step 2: Calculate inputs of the PID Controller
    P(n) = e_tr(n);
    I(n) = integral_e(e_tr);
    D(n) = derivative_e(e_tr);

    %Step 3: Calculate outputs of f (.) nonlinear function
    x_p(n) = f_nonl(e_tr(n),alpha_p,delta_p);
    x_i(n) = f_nonl(I(n),alpha_i,delta_i);
    x_d(n) = f_nonl(D(n),alpha_d,delta_d);

    %Step 4: Calculate Control Signal
    u(n) = K*(w1(n)*x_p(n)+w2(n)*x_i(n)+w3(n)*x_d(n))/(w1(n)+w2(n)+w3(n));

    %Step 5: Apply control signal to the plant and obtain output of the system
    y(n+1) = y_out(u,y,m_paper);

    %Step 6: Update controller parameters
    w1(n+1) = w1(n) + eta1*e_tr(n)*u(n)*x_p(n);
    w2(n+1) = w2(n) + eta2*e_tr(n)*u(n)*x_i(n);
    w3(n+1) = w3(n) + eta3*e_tr(n)*u(n)*x_d(n);

     %Step7:n <-n+1 and go to Step 1.

end
```

Yakup Görür (040130052) HW1 MKM-502 Soft Computing
Assoc.Prof.Dr. Gülay ÖKE GÜNEL