IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 9, November 2014.
www.ijiset.com
ISSN 2348 – 7968

A NEW INTELLIGENT, ROBUST AND SELF TUNED CONTROLLER DESIGN (II).

Farhan A. Salem

Mechatronics engineering program, Department of Mechanical Engineering, College of Engineering, Taif University, 888, Taif, Saudi Arabia. Alpha center for Engineering Studies and Technology Researches, Amman, Jordan. Email:

Abstract
This paper proposes a new, simple, intelligent, robust and self-tuned controller design approach for getting a process under control, while achieving an important design compromise; acceptable stability, and medium fastness of response. The proposed approach is based on analyzing plants step response to calculate it's parameters and based on calculated parameters, calculate controller's parameters, feed it to controller, and repeat process until achieving an acceptable stability, and medium fastness of response. The proposed approach was test using MATLAB m.file and Simulink model for different systems

Keywords: *Controller design, Modeling/Simulation*.

1. Introduction

The term control system design refers to the process of selecting feedback gains (poles and zeros) that meet design specifications in a closed-loop control system. Most design methods are iterative, combining parameter selection with analysis, simulation, and insight into the dynamics of the plant (Katsuhiko Ogata,1997),(Ahmad A. Mahfouz, et al,2013), An important compromise for control system design is to result in acceptable stability, and medium fastness of response, one definition of acceptable stability is when the undershoot that follows the first overshoot of the response is small, or barely observable(Farhan A. Salem,2013), Beside world wide known and applied controllers design method including Ziegler and Nichols known as the "process reaction curve" method (J. G. Ziegler, et al, 1943). and that of Cohen and Coon (G. H. Cohen, 1953) Chiein-Hrones-Reswick (CHR), Wang–Juang–Chan, many controllers design methods have been proposed and can be found in different texts including (Astrom K,J ,et al,1994)(R. Matousek, 2012)(Susmita Das, et al, 2012)(Saeed Tavakoli, et al, 2003)(Astrom K,J, et al, 1994)(Norman S. Nise,2011)(Gene F. Franklin, et ak, 2002)(Dale E. Seborg, et al, 2004), each method has its advantages, and limitations.

This paper extend previous work (Farhan A. Salem,2013)(Farhan A. Salem,2014*)(J. G. Ziegler ,et al,1943)(G. H. Cohen, et al,1953)(Astrom K,J, et al,1994), and proposes a new simple and intelligent controller design approach, that is based on relating controller(s)' parameters and plant's parameters to result in meeting an important design compromise; acceptable stability, and medium fastness of response in terms of minimum PO%, 5T, T_S, and E_{SS}.

2. Proposed approach

(Farhan A. Salem,2013)(Farhan A. Salem,2014*) proposed (P-, PI-, PD- and PID-) controllers design methods with corresponding expressions for calculating controller's parameters $(K_P, K_I$, and $K_D)$, a long with soft tuning parameters $(\alpha, \beta$ and $\varepsilon)$. The proposed methods are based on relating and calculating controller's parameters from plant's parameters (ζ, ω_n, T), to result in overall response with acceptable stability, medium fastness of response and minimum overshoot. For PID controller design, the proposed expressions derived by (Farhan A. Salem,2013)(Farhan A. Salem,2014*) are given in Table 1, where : parameter α is responsible for speeding up response, and reducing Error, meanwhile, ε is responsible for tuning overshoot, where increasing ε will increase overshoot, and vise versa, finally parameter α is responsible for reducing both overshoot and error.

As shown on block diagram representation (Figure 1) ,the proposed method is accomplished as follows; first by subjecting the closed loop system with only proportional controller with gain *set to* $K_P=1$, where K_I and K_D are set to zero , to step input of $R(s)=A/s$, then feeding resulted step response to module with program, that will analyze response curve to calculate plant's parameters *(e.g.* ζ ω_n,*T)*, then use these parameters to assign values to controller's tuning parameters (α, β and ε) and calculate corresponding controller gains (K_P, K_I and K_D) according to Table 1, and then feed calculated controller parameters to controller, to refine the resulted response, the program will subject the overall closed loop system with calculated gains, to the same step input, and feed resulted response to module with program and find corresponding closed loop system parameters and controller's gains, repeating this process to finally calculate controller's parameters, that will result in acceptable stability, medium fastness of response with minimum overshoot, and maintain this response by continuously analyzing resulted response and maintaining it. In case of disturbance input, the proposed method will repeat the mention process, to return the closed loop response to previous state. For example for second order systems, the program will use system's step response to calculate system's damping ratio ζ , undamped natural frequency ω_n, and time constant T. for higher order systems, the program to use system's step response to analyze response and apply dominant poles approximation, and find corresponding plants approximated parameters, and proceed as mentioned.

Table 1: Proposed expressions for PID parameters calculation

Plant		PID parameters					
		K_P	K_I	K_D	T_D	T_I	N
ζ	ω_n	β	$\frac{\omega_n}{2\xi}$	$\frac{1}{2\xi\omega_n}$	$\frac{1}{2\xi\omega_n}$	$\frac{2\xi}{\omega_n}$	2 ÷ 20
Tuning limits		0:inf	$\varepsilon\frac{\omega_n}{2\xi}$, $\varepsilon = 0.1 \div 10$	$\alpha\frac{1}{2\xi\omega_n}$, $\alpha = 0.58 \div 1.5$	$\frac{\alpha}{2\xi\omega_n}$	$\frac{2\xi}{\varepsilon\omega_n}$	2 ÷ 20

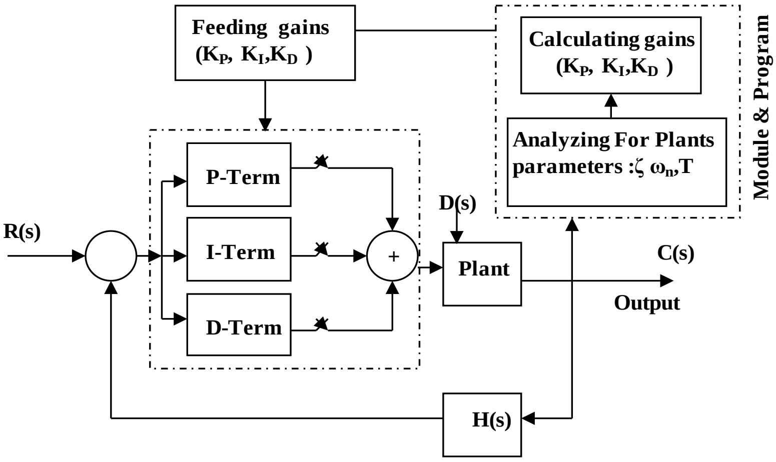

Figure 1 Block diagram representation of the proposed approach

2. Testing proposed approach

To clarify the operation of the proposed method, MATLAB/Simulink will be used to simulate the proposed method, write program m.file for response analysis, calculating plant's parameters, calculating controller parameters, feeding calculated gains to controller, subjecting overall closed loop system with calculated gains to step input, and repeating the process until reaching, acceptable stability, medium fastness of response and minimum overshoot . To test the proposed approach, Simulink model shown in Figure 2 is developed , this model is with four plants of different orders, including PMDC motor system as prime mover to be used for both, mobile robot speed control, and robot arm position control.

2.1 Testing for mobile robot linear speed control

The DC motor open loop transfer function without load attached relating the input voltage, $V_{in}(s)$, to the motor shaft output angular speed is given by Eq.(1). To model, Simulate and analyze the open loop plant system, the total equivalent inertia, J_{equiv} and total equivalent damping, b_{equiv} at the armature of the motor are given by Eq.(2). Since, the geometry of the mechanical part determines the moment of inertia, to compute the total inertia, J_{equiv} mobile platform, system can be considered to be of cylindrical shape ⊏, with the inertia calculated as given by Eq.(3). To calculate the load torque of mobile platform, the following forces can be included; the hill-climbing resistance force F_{climb}, aerodynamic Drag force , F_{aerod} and the linear acceleration force F_{acc}, as given by Eq.(4), with their corresponding torques. The mobile robot has the following nominal values; M=50 Kg, height , h=1 M, width, a= 0.8 M , the wheel radius is to be 0.075. Figure 2(c) shows Simulink model of Mobile robot torque (Farhan A. Salem,2013*)(Farhan A. Salem,2014*). The tachometer constant is selected as given by Eq.(5), to result in linear velocity of 0.5 m/s for 12 input voltage . The following nominal values for the various parameters of eclectic DC motor used: V_{in}=12 *Volts;* J_m= 0.271 *kg·m²;* b_m = 0.271;K_t = 1.1882 *N-m/A;* K_b = 1.185 *V-s/rad;* R_a =0.1557 *Ohm;* L_a=0.82*Henry*. Based on Eqs.(1,5), proposed approach, and refereeing to(Farhan A. Salem,2013*)(Farhan A. Salem,2014*)(Ahmad A. Mahfouz, et al, 2013)(*-Farhan A. Salem(2013). the Simulink model consisting of sub-models shown in Figure 2 are developed to be used to test proposed approach.

$$G_{speed}(s) = \frac{\omega(s)}{V_{in}(s)} = \frac{K_t / n}{L_a J_m s^2 + (R_a J_m + b_m L_a)s + (R_a b_m + K_t K_b)} \quad (1)$$

$$b_{equiv} = b_m + b_{Load}\left(\frac{N_1}{N_2}\right)^2 \quad , \quad J_{equiv} = J_m + J_{Load}\left(\frac{N_1}{N_2}\right)^2 \quad (2)$$

$$J_{load} = \frac{bh^3}{12} \Leftrightarrow J_{equiv} = J_{motor} + J_{gear} + (J_{wheel} + mr^2)\left(\frac{N_1}{N_2}\right)^2 \quad (3)$$

$$F = 0.5\rho A C_D \upsilon^2 + Mg\sin(\alpha) + m\frac{d\upsilon}{dt} \quad (4)$$

$$K_{tach} = \frac{V_{in}\max}{\omega_{\max}} = \frac{V_{in}\max}{\upsilon_{\max}/\mathrm{r}} = \frac{12}{0.5/0.75} = 6.667 \quad (5)$$

ISSN 2348 – 7968

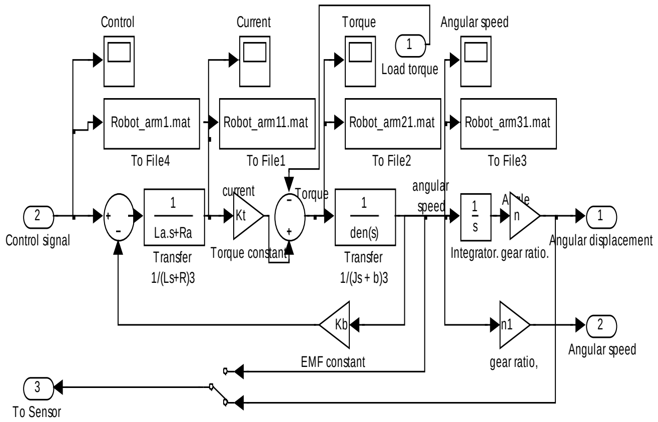

Figure 2 (a) DC machine subsystem model for mobile robot and robot arm

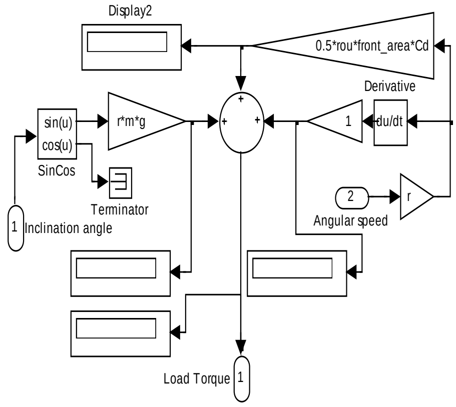

Figure 2 (c) Mobile robot torque

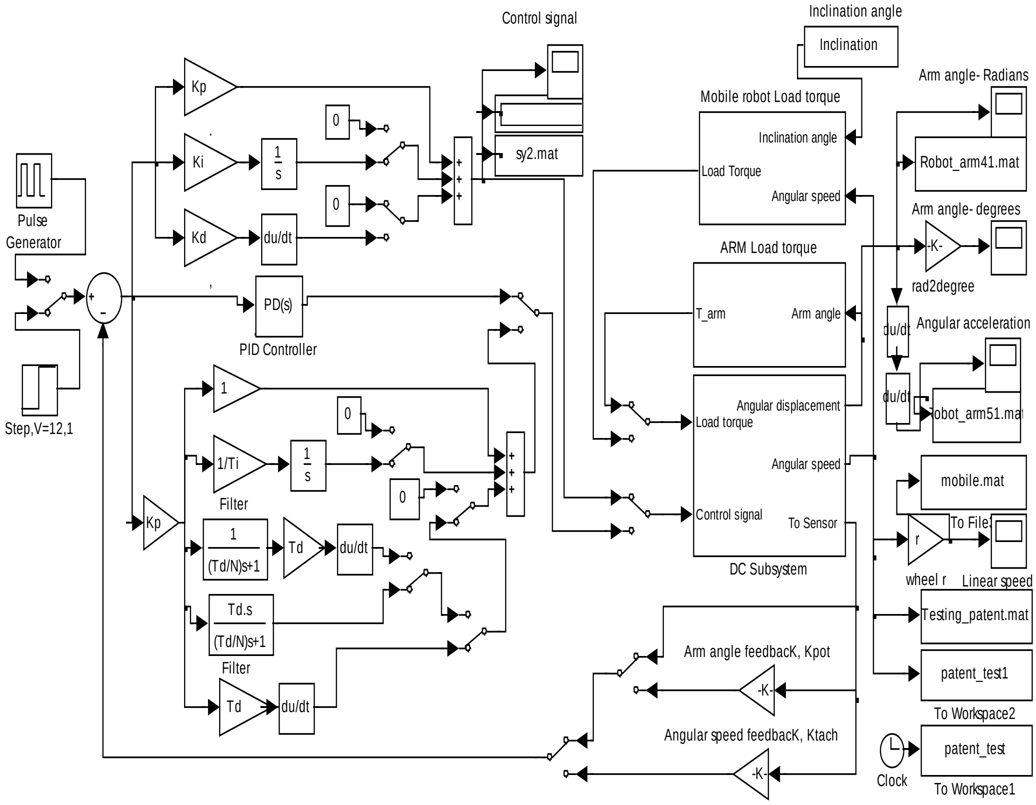

Figure 2 (d) Simulink model used to test proposed approach.

ISSN 2348 – 7968

Switching Simulink model to mobile robot speed control, Running program (m.file), for controlling the output linear speed of mobile robot to be 0.5 m/s for input of 12V. First, the input to the program, is the resulted closed loop response curve of overall closed loop system with only proportional gain set to $K_P=1$, and K_I and K_D are set to zero, program will analyze the response curve and calculate plant's parameters, response measures (listed in Table 2(a)) approximated second order transfer function (given by Eq.(6)), and plots both original and calculated response (shown Figure 3(a)), and feed calculated gains to PID controller, then the program will subject the closed loop system with PID controller with calculated gains, to the same step input, and then , program will take as input resulted response curve, analyze it, to calculate overall closed loop approximated second order system, calculate controller gains and calculate tuning parameters (listed in Table 2) and feed it to controller, and then, once again run the simulation and repeat the analysis, calculation and feeding process, if acceptable response is resulted, maintain it, if not, calculate the new controller gains based on new closed loop response curve analysis, repeat the process untill acceptable response is achieved For *second* program self-run , with the calculated gains, the resulted response curve is shown in Figure 3(b), and overall system parameters are listed in Table 2(b), approximated closed loop transfer function is given by Eq.(7).

For *third* program self-run , with second calculated gains,the resulted resp ... all system parameters

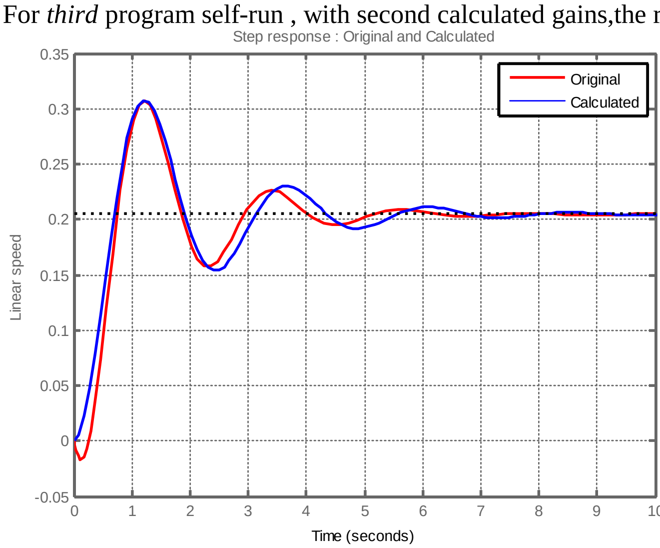

Figure 3(a) first program self- run: plots of both original and calculated step response

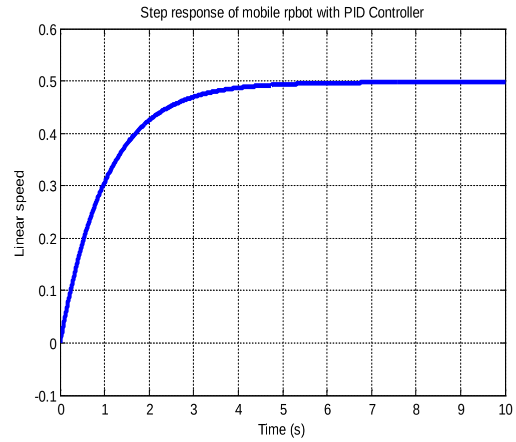

Figure 3(b) Second program self- run: Step response of mobile robot with PID controller

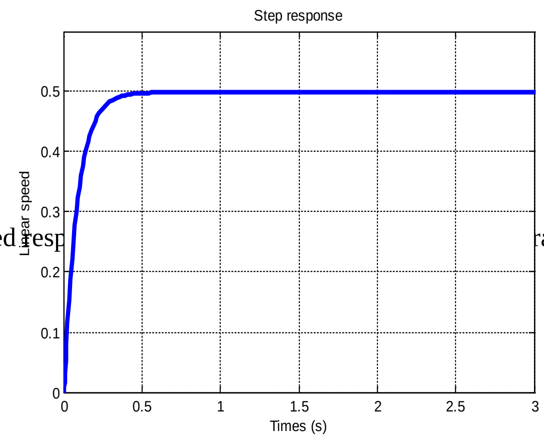

Figure 3(b) Third program self- run: Step response of mobile robot with PID controller

Table 2(a): calculated system parameters, , response measures, first run

Original mobile robot system	Calculated parameters	**ε**	**ω_n**	**T_R**	**M_P**	**T_P**	**Peak value**	**PO%**	**DC gain**
		0.215434	2.63921	0.804904	0.102384	1.21898	4.09518	0.500034	0.204754
	Calculated gain	**β**	**ε**	**α**	**K_P**	**K_I**	**K_D**		
		80	95	1	80	0.284287	83.5423		

Table 2(b): Calculated system parameters, response measures, second self-run with calculated gain

Original mobile robot system	Calculated Plant's parameters	ε	ω_n	T_R	M_P	T_P	Peak value	PO%	DC gain
		0.95901	1.13037	3.1263	1.20e-05	-	0.497525	2.413e-05	0.497513
	Calculated Controller's parameters	β	ε	α	K_P	K_I	K_D		
		100	60	1	60	0.542016	27.6744		

Table 2(c): Calculated system parameters, response measures, Third self-run with calculated gain

Original mobile robot system	Calculated Plant's parameters	ε	ω_n	T_R	M_P	T_P	Peak value	PO%	DC gain
		0.9168	0.839	0.267	36.8e-5	-	0.499313	73.786e-5	0.498945
	Calculated Controller's parameters	β	ε	α	K_P	K_I	K_D		
		300	60	1	100	0.448566	38.9669		

$$G(s)=\frac{1.426}{s^2+1.137s+6.965}$$ Eq.(6)

$$G(s)=\frac{0.6357}{s^2+2.168s+1.278}$$ Eq.(7)

$$G(s)=\frac{0.3519}{s^2+1.54s+0.7052}$$ Eq.(8)

2.2 Testing proposed approach for different systems

Case (1): Testing the proposed approach for third order system with transfer function given by Eq.(9), for desired output of 12, for step input of R(s)=12/s, will result in approximated second order transfer function given by Eq.(10), and calculated plants and PID parameters and response measures, shown in Table 5(a), both response plots of original and compensated

controlled systems are shown in Figure 4(a). The calculated plants parameters are used by program to calculate controller's parameters, feed it to controller and subject system to step input , and repeat process. For third program self-run , with calculated gains, will result in response curve shown in Figure 4(b), and overall system parameters listed in Table 2(b), with approximated closed loop transfer function given by Eq.(11), and final closed loop response shown in Figure 4(b), this is an acceptable response , that the program will maintain, for each next run, and return the response to it in case of disturbance input.

$$G(s)=\frac{s+1}{s^3+3s^2+s}$$ Eq.(9)

$$G(s)=\frac{13.58}{s^2+1.074s+1.132}$$ Eq.(10)

Table 5(a): calculated system parameters, , response measures, first run

Original mobile robot system	Calculated Plant's parameters	ε	ω_n	T_R	M_P	T_P	Peak value	PO%	DC gain
		0.660966	0.926658	3.38803	0.00521428	12.6443	12.0438	0.00043313	12.0386
	Calculated Controller's parameters	β	ε	α	K_P	K_I	K_D		
		80	95	1	80	0.268534	88.4431		

Table 5(b): calculated system parameters, , response measures, first run

Original mobile robot system	Calculated Plant's parameters	ε	ω_n	T_R	M_P	T_P	Peak value	PO%	DC gain
		0.894306	2.22795	0.5164	0.0225299	3.15134	12.036	0.00187538	12.0135
	Calculated Controller's parameters	β	ε	α	K_P	K_I	K_D		
		75	60	1	75	0.996234	15.0567		

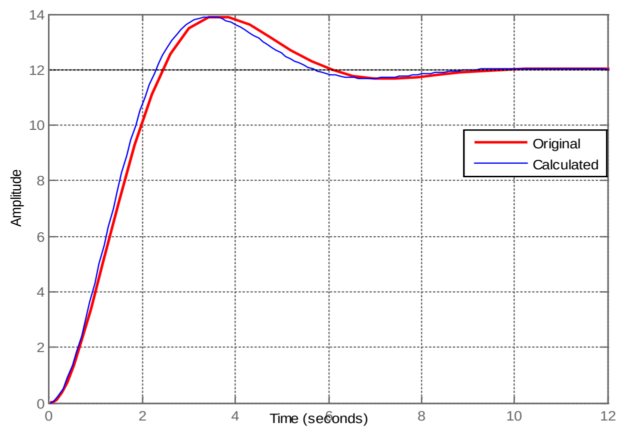

Figure 4(a) response plots of original and approximated systems

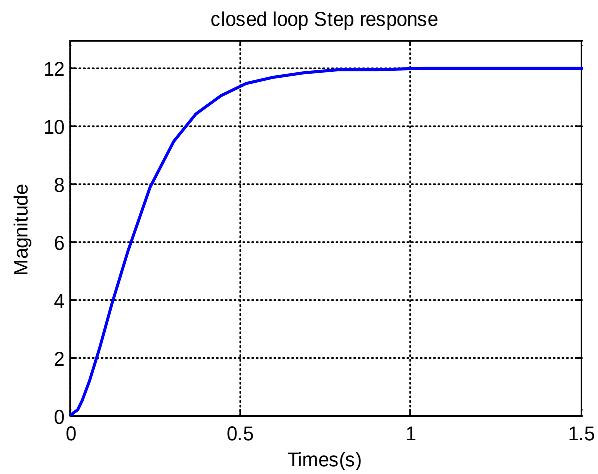

Figure 4(b) closed loop response with PID

Case (2): Testing the proposed approach for fourth order system with transfer function given by Eq.(12), for desired output of 12, for step input of R(s)=12/s, will result in approximated second order transfer function given by Eq.(13), and calculated plants parameters, PID parameters and response measures, shown in Table 5(a), both response plots of original and compensated systems are shown in Figure 5(a). The calculated plants parameters are used by program to calculate controller's parameters , feed it to controller and subject system to step input , and repeat process, until an acceptable response is achieve (shown in Figure 5(b)).

$$G(s)=\frac{s^2+5s+3}{10s^4+2s^3+20s^2+5s+1}$$

Eq.(12)

$$G(s)=\frac{3.379}{s^2+0.2882s+0.3754}$$

Eq.(13)

Table 6(a): calculated system parameters, , response measures, first run

Original mobile robot system	Calculated Plant's parameters	ε	ω_n	T_R	M_P	T_P	Peak value	PO%	DC gain
		0.235194	0.612696	3.53238	4.20857	5.27547	13.2095	0.46757	9.00094
	Calculated Controller's parameters	β	ε	α	K_P	K_I	K_D		
		140	60	1	140	0.123748	121.214		

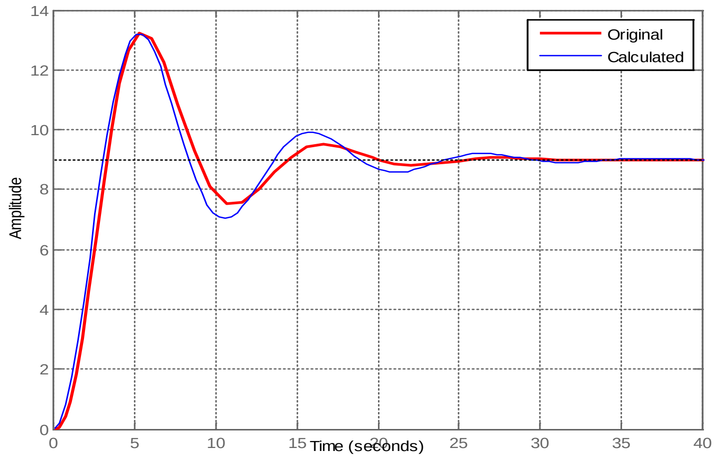

Figure 5(a) response plots of original and approximated systems

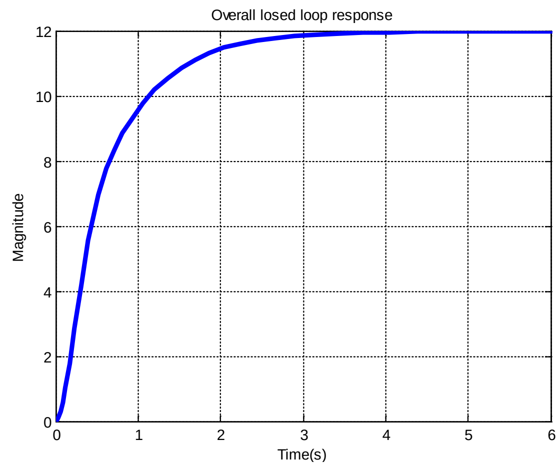

Figure 5(b) closed loop response with PID

2.3 Program code

```
Kp=1;Ki=0;Kd=0;
while(1)
  clc,close all
  format bank
  disp('====================================')
  disp( '        Testing Response Data ')
  disp( '        ')
  disp('====================================')
plot(patent_test,patent_test1,'r');
figure
plot(patent_test,patent_test1,'r');
response_data=[patent_test1 ,patent_test ]';
if     any(response_data(1,[20:end])<0);%     undamped
response
Kp=1;Kd=5 ;Ki=15;%
else
 [a,b]=max(response_data(1,:));
final_steady_satae_value=response_data(1,end);
fprintf('    final    steady    satae    value    =    %g
\n',final_steady_satae_value)
ww=response_data(1,:);
for z0= 1:length(response_data(1,:))
if response_data(1,z0)>= final_steady_satae_value*.95
   rise_time=response_data(2,z0);
   break, end, end
fprintf(' Rise time = %g  \n',rise_time)
overshoot= a-final_steady_satae_value;
fprintf(' Overshoot = %g  \n',overshoot)
if (overshoot <= 0)==1
  ZETA1= 1;
  omega_n=(4.67*ZETA1-1.2)/rise_time;
  settling_time=(9.31*ZETA1-3.4)/omega_n
 Percent_overshoot=0;
fprintf('  Undamped  natural  frequency,  omega_n  =  %g
\n',omega_n);
fprintf(' Damping ratio, ZETA1 = %g  \n',ZETA1)
%=================
alpha=10;
%=================
else
 Peak_value= a ;
fprintf(' Peak Value = %g  \n',a)
Peak_time=response_data(2,b);
fprintf(' Peak Time = %g  \n',Peak_time)
Percent_overshoot=((Peak_value-
final_steady_satae_value)/final_steady_satae_value);%
*100;
fprintf(' Overshoot = %g  \n',overshoot)
fprintf(' Percent overshoot = %g  \n',Percent_overshoot)
disp('
==============================================
=========================')
ZETA1=-log(Percent_overshoot)
/(sqrt((log(Percent_overshoot))^2+ (pi)^2));
fprintf(' Damping ratio, ZETA1 = %g  \n',ZETA1)
omega_n=pi/(Peak_time*sqrt(1-ZETA1^2))  ;%calculating
from peak time
fprintf('  Undamped  natural  frequency,  omega_n  =  %g
\n',omega_n);end
transfer_function=tf(final_steady_satae_value*omega_n^2,
[1 , 2*ZETA1*omega_n , omega_n^2])
hold, step(transfer_function), grid
legend('Original', 'Calculated'), end
if (Percent_overshoot > 0.7)
 beta=80;epsilon=100;alpha=1;
elseif (Percent_overshoot > 0.1)&&(Percent_overshoot <=
0.7)
 beta=80;epsilon=95;%80, 85, alpha=1;
elseif(Percent_overshoot < 0.1)&&(Percent_overshoot >=
0.05)
beta=80;epsilon=70;alpha=1;
elseif(Percent_overshoot < 0.05)  %< 0.05
 beta=140%200, epsilon=60;alpha=1;end
Kp=beta ;% Speeds up response
Kd=(1/(2*ZETA1*omega_n))*     epsilon;%     Reduces
overshoot
Ki=(omega_n/2*ZETA1)*alpha;%
disp('====================================')
fprintf('  alpha= %g  \n',alpha)
fprintf('  epsilon= %g  \n',epsilon)
fprintf('  Kp = %g  \n',Kp)
fprintf('  Ki = %g  \n',Ki)
fprintf('  Kd = %g  \n',Kd)
pause(3)
%===========================
sim( 'patent_test_1_1_1')% runing siulink model entitled
patent_test_1_1_1
%===========================
pause(10)
end
```

Conclusion

A new, simple, intelligent, robust and self-tuned controller design approach for getting a process under control, while achieving an important design compromise is proposed and tested. the proposed approach is based on relating controller(s)' parameters and plant's parameters to result in meeting an important design compromise; acceptable stability, and medium fastness of response in terms of minimum *PO%*, *5T*, T_S, and E_{SS}.. The proposed approach was test using MATLAB m.file and Simulink model for different systems, the theoretical results show the simplicity and applicability of the proposed approach.

As future work, to modify program, such that it can be applied for most types of systems, and to suggest and built a circuit design to read and analyze plant's response parameters.

References

Katsuhiko Ogata (1997), modern control engineering, third edition, Prentice hall, .

Ahmad A. Mahfouz (2013), Mohammed M. K., Farhan A. Salem, "Modeling, Simulation and Dynamics Analysis Issues of Electric Motor, for Mechatronics Applications, Using Different Approaches and Verification by MATLAB/Simulink", IJISA, vol.5, no.5, pp.39-57.

Farhan A. Salem(2013), , New controllers efficient model-based design method, European Scientific Journal May ed. vol.9, No.15.

J. G. Ziegler (1943), N. B. Nichols. "Process Lags in Automatic Control Circuits", Trans. ASME, 65, pp. 433-444.

G. H. Cohen (1953), G. A. Coon. "Theoretical Consideration of Related Control", Trans. ASME, 75, pp. 827-834.

Astrom K,J (1994), T. Hagllund, PID controllers Theory, Design and Tuning , 2nd edition, Instrument Society of America,

R. Matousek (2012), HC12: Efficient PID Controller Design, Engineering Letters, pp 41-48, 20:1,

Susmita Das(2012), Ayan Chakraborty,, Jayanta Kumar Ray, Soumyendu Bhattacharjee. Biswarup Neogi, Study on Different Tuning Approach with Incorporation of Simulation Aspect for Z-N (Ziegler-Nichols) Rules, International Journal of Scientific and Research Publications, Volume 2, Issue 8, August

Saeed Tavakoli (2003), Mahdi Tavakoli, optimal tuning of PID controllers for first order plus time delay models using dimensional analysis, The Fourth International Conference on Control and Automation (ICCA'03), 10-12 June , Montreal, Canada.

Astrom K,J (1994), T. Hagllund, PID controllers Theory, Design and Tuning , 2nd edition, Instrument Society of America,

Norman S. Nise(2011), Control system engineering, Sixth Edition John Wiley & Sons, Inc,

Gene F. Franklin (2002), J. David Powell, and Abbas Emami-Naeini, Feedback Control of Dynamic Systems, 4th Ed., Prentice Hall, .

Dale E. Seborg(2004), Thomas F. Edgar, Duncan A. Mellichamp ,Process dynamics and control, Second edition, Wiley

*Farhan A. Salem (2014) ' New controllers' design method for first order systems and systems that can be approximated as first order' .

*Farhan A. Salem(2014) ' New efficient controllers' design method for second order systems and systems that can be approximated as SECOND order' .

*Farhan A. Salem (2014) ' New Controllers design method for FOPDT/SOPDT processes' International Journal of Engineering Research And Management (IJERM) ISSN : 2349- 2058, Volume-01, Issue-07, October 2014

*Farhan A. Salem(2013) , Dynamic and Kinematic Models and Control for Differential Drive Mobile Robots, International Journal of Current Engineering and Technology, Vol.3, No.2 , PP253-263,June

**Farhan A. Salem (2013), Refined models and control solutions for Mechatronics design of mobile robotic platforms, Estonian Journal of Engineering, 2013, 19, 3, 212–238

**Farhan A. Salem,(2013) Dynamic Modeling, Simulation and Control of Electric Machines for Mechatronics Applications, int. journal of control, automation and systems, vol.1 No.2, pp33-42, April 2013

Ahmad A. Mahfouz(2013), Mohammed M. K., Farhan A. Salem, Modeling, Simulation and Dynamics Analysis Issues of Electric Motor, for Mechatronics Applications, Using Different Approaches and Verification by MATLAB/Simulink (I). I.J. Intelligent Systems and Applications, 2013, 05, 39-57

*-Farhan A. Salem(2013), Mechatronics Design of Motion Systems; Modeling, Control and Verification. International Journal of Mechanical & Mechatronics Engineering IJMME-IJENS Vol:13 No:02 , pp1-17.

Farhan A. Salem: Now with Taif University, college of engineering, Dept. of mechanical Engineering, Mechatronics engineering.