COMPUTATIONAL INTELLIGENCE (CI) SELF-ADAPTIVE PID (CISAPID)

Mingzhong Yi Yong Qin LiMin Jia

Institute of Computing Technologies Head, Res. Centre for Intelligent Systems
CHINA ACADEMY OF RAILWAY SCIENCES

Abstract: A new method for Computational Intelligence (CI), CI formulated with Analytic Functions and Logics, is given, and a new PID controller - CISAPID is put forward. CISAPID take the strategy attempt to incarnate but not to realize or imitate the complex thinking and behaviour of human being. The constitution, principle and qualitative arguments tuning experience of CISAPID are analysed in detail. Simulation and practical application show that the performance of CISAPID is better than that of general PID.

Keywords: CI AI incarnate CISAPID PID

1. PREFACE

Computational Intelligence (CI) is a new chapter in Intelligence Theory[3]. The reason why the research of Artificial Intelligence (AI) did not achieve expected progression perhaps is that AI depend on excessively the advantage of computer - precise arithmetic and the fast calculation to imitate the complex thinking and action of human being[8]. But thinking and action of human being do not often depend on precise arithmetic and fast calculation, furthermore, the ability to abstract of human being plays, I think, the most important role in the intelligence of human being, but this ability is just what computer lacks extremely. So, an one-year old babe perhaps don't understand any arithmetic, and can't calculate at all, but some intelligence of his or hers is far more better than any current sophisticated computer. It is possible that the thinking mechanism of human being can hardly be cryptanalyzed completely, so, what machine can achieve or imitate is perhaps only fragmentary and dyshematopietic intelligence of human being forever, no matter what advanced machine can hardly reproduce entirely the complicated mechanism of the thinking of human being[7] [8].

But for practical application in engineering, it is enough to achieve our given purpose, to obtain a satisfactory but not optimal result, and the truly advanced theory, algorithm or method perhaps are those that are not complicated, sophisticated and optimal but practical, feasible, satisfactory, reliable and simple, thus, take a strategy attempt to incarnate but not to realize or imitate the complex thinking and behavior of human being is not only a remedy for computer's shortcoming, but also a practical, feasible and simple shortcut. So, in this paper, a new method for CI - CI formulated with Analytic Functions and Logics is given, Logics mentioned in this paper are expressed as "if … , then … " instructions of computer program, they are enough to incarnate logical relation for practical project, it is no need to resort to more complex method.

2. COMBINE CI WITH PID

Today, the most popular controller is still PID controller, even in developed country - Japan, the rate of utilization of PID controller reached 84.5%[10].

But general PID has many intrinsic shortcomings. In order to improve the performance of general PID , many scholars had done a lot of research. So, how to combine Computational Intelligence with widely used general PID is very significative in theory and in practice. So, in this paper, a new PID controller-CISAPID is put forward.

In our practical project for engineering, this controller seems not highly depend on the precise model of controlled object (we could not find the precise model), but make use of some Analytic Functions and Logics to regulate arguments of PID real-timely according to the feedback information, to incarnate some intelligence of human being to a certain degree in a practical, flexible, simple way, and to incarnate knowledge, experience and rules of experts and skilled operators, thus, it has some characteristics of Fuzzy Control and Expert Control. But it is not confined to the patterns of Fuzzy Control and Expert System, because fuzzy rule table is not intuitionistic, and is not convenient to establish, furthermore, we don't want to make the control system too complex. In practical project, the more simple a system is, the more practical, feasible, reliable and robust it often is. So, if it is able to achieve the same satisfactory efficiency, the control system should be simple as best as possible, thus, not only the control efficiency is improved, but also the hardware cost and the developing cycle are reduced markedly, thus this system is advanced in fact. According to the viewpoint of James C. Bezdek, CI is based on the data provided by operator, but traditional AI is on so-called "knowledge". He defined the CI system as follows: When a system only treats with the data from bottom, and possess the part for pattern recognition, and don't make use of knowledge in the sense of AI, then, this system can be viewed as a CI system. Such a system would have characteristics as follows: has the adaptability of compute; has the tolerance of compute error; close to the speed that human handling problem; close to the error rate of human being[3]. So, we can also think in the same way: If a controller treats with the data from bottom only based on Logics and Analytic Functions abstracted from experience, rules and knowledge of experts and operators, and possess the part for pattern recognition, and don't make use of knowledge in the sense of AI, then this controller can also be viewed as a CI controller. So, CISAPID can also be viewed as a controller based on CI.

3. CISAPID

For typical standard negative feedback control system, general PID controller can be expressed as:

$$u=K_p(e+\frac{1}{T_i}\int e\,dt+T_d\frac{de}{dt}) \tag{1}$$

The CI Self-adaptive PID Controller (CISAPID) can be formulated as follows[6] [5]:

$$u=(1-k\_u)*(Kp\_e*e + Ki\_e*\int e\,dt + Kd\_e\_de*\frac{de}{dt} + Kd\_dde*\frac{d^2e}{dt^2}) + k\_u*u_0 \tag{2}$$

$$Kp\_e=Kp0\_e+Kp1\_e*(1-exp(-Kp\_s*W\_Kp\_e*(e-p*s)^2)) \tag{3}$$
$$Ki\_e=Ki0\_e+Ki1\_e*exp(-W\_Ki\_e*e^2) \tag{4}$$
$$Kd\_e\_de=Kd\_e+Kd\_de \tag{5}$$
$$Kd\_e=Kd0\_e+Kd1\_e*exp(-Kd\_s*W\_Kd\_e*(e-p*s)^2) \tag{6}$$
$$Kd\_de=Kd0\_de+Kd1\_de*exp(-W\_Kd\_de*(de)^2) \tag{7}$$
$$Kd\_dde=Kd0\_dde+Kd1\_dde*exp(-W\_Kd\_dde*(d^2e)^2) \tag{8}$$
$$u_0=(u\_0+u\_1*u\_power+u\_2*(u\_power)^2+...+u\_n*(u\_power)^n+...)/(1+u\_power+(u\_power)^2+...+(u\_power)^n+...) \tag{9}$$

e=setting value-actual value, error *e*. *Kp_e*, *Ki_e*, *Kd_e* indicate that these arguments are related to *e*. *Kd_de* indicate that the argument is related to *de*, the first-order differential of *e*; *Kd_e_de* indicate that the argument is related to *e* and *de*; *Kd_dde* indicate that the argument is related to d^2e, the second-order differential of *e*; So as to the rest. If arguments except *Kp0_e*, *Ki0_e*, *Kd0_e* are all 0, then, CISAPID turn itself back to general PID. The reasons why we construct Analytic Functions as above and more detailed information about arguments tuning, please refer to my master's degree thesis. The tuning method of *Kp0_e*, *Ki0_e*, *Kd0_e* can refer to the tuning method of general PID based on object model or dynamic response curve[1], such as Ziegler Nichols - frequency response method[9], CohenCoon - response curve method[2],integral squared error - ISE[4] and so on. Because physical meaning of the other arguments are explicit, simple, and regular, so, it is not very difficult to determine them by off-line simulation or resort to experience and by means of trial-and-error method. Further more, what needed to tune are their initial arguments, the running arguments are self-adjust online and real-timely based on the initial arguments according the situation on-site. Even if you did not tune these arguments very well, or the controlled object and other factors had already changed, the control efficiency would not decline greatly (but the burden of executing mechanism would perhaps increase), thus, the self-adaptability and robust of this controller are good.

3.1 The proportional action of CISAPID

$$Kp\_e=Kp0\_e+Kp1\_e*(1-exp(-Kp\_s*W\_Kp\_e*(e-p*s)^2)) \tag{3}$$

Kp_e is similar to a reversed double peak gaussian function, the larger the *W_Kp_e* is, the more sharp the curve is, the value of *W_Kp_e* should ensure that system would respond enough proportional action within a wide range, so the value of *W_Kp_e* should be minor; *p* is a sign variable, when $e<0$, $p=-1$; when $e\geqslant 0$, $p=+1$. $s\geqslant 0$ is an offset; *Kp_s* is related to offset *s*, and is a coefficient to adjust *W_Kp_e*, when $|e|\geqslant s$, $Kp\_s=1$, when $|e|<s$, it is allowed $Kp\_s\neq 1$. Because of offset *s*, the minimum value of *Kp_e* is not at the point of $e=0$, but at the point of $e=+s$ or $-s$. If system is not very stable, and the requirement for accuracy and rapidity are not high when system is near to the equilibrium point, then, *Kp_s* should be nearly equal to 0, thus, when $|e|<s$, *Kp_e* would hardly increase with the reduction of $|e|$, and seems to be a constant, and this is beneficial to system stability.
Analytic Function of formula (3) in fact incarnated knowledge, rules and experience of experts and operators, and approximately incarnated the fuzzy rules as follows:
if |*e*| is "extreme big", **then** proportional action *Kp_e* should be "very big";
if |*e*| is "very big", **then** proportional action *Kp_e* should be "comparatively big";
if |*e*| is "comparatively big", **then** proportional action *Kp_e* should be "not big and not small";
if |*e*| is "a bit big" (namely, |*e*| is a little bigger than *s*), **then** proportional action *Kp_e* should be "comparatively small";
if |*e*| is "not big and not small" (namely, |*e*|=*s*), **then** proportional action *Kp_e* should be "minimum";
if |*e*| is "comparatively small" (namely, |*e*| is a little smaller than *s*), **then** proportional action *Kp_e* should be "comparatively small";
if |*e*| is "very small" (namely, |*e*| is approaching to 0 or |*e*|=0), **then** proportional action *Kp_e* should be "not big and not small";

3.2 The integral action of CISAPID

$$Ki\_e=Ki0\_e+Ki1\_e*exp(-W\_Ki\_e*e^2) \tag{4}$$

Ki_e is a gaussian function related to error *e*. The value of *W_Ki_e* should cause system respond integral action only within narrow ranges (error is very small). When error *e* becomes a little big, integral action should be near to 0 in order to carry out the isolation of integral action and to avoid integral saturation. So the value of *W_Ki_e* should be a little larger, then the curve of *Ki_e* would be very sharp. $Ki0\_e$ should be far less than $Ki1\_e$ to help to realize the isolation of integral.

Analytic Function of formula (4) in fact incarnated knowledge, rules and experience of experts and operators, and approximately incarnated the fuzzy rules as follows:

if |*e*| is "extreme big", **then** integral action Ki_e should be "extreme small";

if |*e*| is "very big", **then** integral action Ki_e should be "extreme small";

if |*e*| is "comparatively big", **then** integral action Ki_e should be "very small";

if |*e*| is "a bit big" (namely, |*e*| is a little bigger than *s*), **then** integral action Ki_e should be "very small";

if |*e*| is "not big and not small", **then** integral action Ki_e should be "comparatively small";

if |*e*| is "comparatively small", **then** integral action Ki_e should be "a bit small";

if |*e*| is "very small" (namely, |*e*| is approaching to 0 or |*e*|=0), **then** integral action Ki_e should be "not big and not small";

3.3 The differential action of CISAPID

$$Kd\_e\_de=Kd\_e+ Kd\_de \quad (5)$$

$$Kd\_e=Kd0\_e+Kd1\_e*exp(-Kd\_s*W\_Kd\_e*(e-p*s)^2) \quad (6)$$

$$Kd\_de= Kd0\_de+ Kd1\_de*exp(-W\_Kd\_de*(de)^2) \quad (7)$$

$$Kd\_dde= Kd0\_dde+ Kd1\_dde*exp(-W\_Kd\_dde*(d^2e)^2) \quad (8)$$

Kd_e is a double peak gaussian function. *p* is a sign variable, when $e<0, p=-1$;when $e\geqslant 0, p=+1$. $s\geqslant 0$ is an offset, and is not the same value as that of formula (3); Kd_s is related to *s*, and is a coefficient to adjust W_Kd_e, when $|e|\geqslant s$, $Kd\_s=1$, when $|e|<s$, it is allowed that $Kd\_s\neq 1$. For the reason of simple, you can assign $Kd\_s=1$. But if disturbance is severe, then, you should assign $Kd\_s>1$, then, when |*e*| is approaching to *0*, differential action Kd_e can reduce more quickly, thus, system would not be very sensitive to disturbance.

Analytic Function of formula (6) in fact incarnated knowledge, rules and experience of experts and operator, incarnated approximately the fuzzy rules as follows:

if |*e*| is "extreme big", **then** differential action Kd_e should be "extreme small";

if |*e*| is "very big", **then** differential action Kd_e should be "extreme small";

if |*e*| is "comparatively big", **then** differential action Kd_e should be "comparatively small";

if |*e*| is "a bit big" (namely, |*e*| is a little bigger than *s*), **then** differential action Kd_e should be "not big and not small";

if |*e*| is "not big and not small" (namely, |*e*|=*s*), **then** differential action Kd_e should be "maximum";

if |*e*| is "comparatively small" (namely, |*e*| is a little smaller than *s*), **then** differential action Kd_e should be "comparatively big";

if |*e*| is "very small" (namely, |*e*| is approaching to 0 or |*e*|=0), **then** differential action Kd_e should be "not big and not small";

Kd_de and Kd_dde are help to control more ahead when controlled object is of very great inertia and hysteresis such as furnace temperature control system. If the inertia and hysteresis are not very great, or if filtering for *de* and *dde* are not very satisfactory, then Kd_de and Kd_dde are not necessary. Knowledge, rules, experience of experts and operators and the fuzzy rules incarnated by the Analytic Functions of formula (7)(8) are similar to those of (3), the main deffrence is that the curve of Ki_e should be very sharp.

From above, we can see that CISAPID is not only a controller of proportional action, integral action and differential action, it is actually related to the first-order and second-order differential of error *e*, the ability that it can control ahead according to the error tendency is very strong, so, engineering application of this thesis is kiln temperature control system with very great inertia and hysteresis.

3.4 Dynamical Weighting Average Algorithm with selection

It is also important to make use of u_0. When adjust system on-site, because of the intrinsic shortcoming of general PID and complexity of control system, you can only compromise among stability, rapidity, accuracy and anti-disturbance. System would often oscillate even if you make great effort to tune the arguments of PID. The purpose of u_0 is just to turn this disadvantage into advantage.

It can be observed, the oscillation of output *u* is often symmetrical about "a specific value", so, the oscillation in fact provide some important information: this "a specific value" is probably close to so-called "setting value" or "right value", so, if we can properly figure out this "a specific value" and let it be u_0, and add this u_0 to output *u*, then, it is equivalent to give a reference point to output *u* (if k_u=0.5), and it also seems to calculate output *u* based on "setting value" or "right value", thus, it is possible that output *u* would probably be just right, and thus system would eliminate oscillate very soon of its own accord. the calculation of u_0 is as follows:

$$u_0=(u\_0+u\_1*u\_power+u\_2*(u\_power)^2+...+u\_n*(u\_power)^n+...)/(1+u\_power+(u\_power)^2+...+(u\_power)^n+...) \quad (9)$$

u_0, u_1, u_2, ... , u_n, ... represent the value of output *u* at current moment, previous one moment, previous two moment, … , previous *n* moment, … , This is in fact the Dynamical Weighting Average of output *u* at each moment. The selection of weight coefficient u_power is very important, if we make u_power=0.9847, then, u_300 would have little effect on u_0, because u_300 is multiplied a coefficient $u\_power^{300}=0.9847^{300}=0.009799<0.01=1\%$. So, the nearer u_i approach to current moment, the more effects it has on u_0; the further u_i is away from current moment, the less effects it has on u_0. This Dynamical Weighting Average Algorithm is coincide with practical situation, the u_0 that figured out as above mainly reflect current working information, but also reflect previous working information to certain extent, thus, this u_0 is quite possible to close to so-called "setting value" or "right value". Further more, we can also multiply a coefficient u_power_i before corresponding u_i.

u_power_i=Ku0_e/(Ku0_e+Ku1_e)+Ku1_e/(Ku0_e +Ku1_e)*exp(-W_Ku_e*(e_i)^2) (10)

Assign biggish value to *W_Ku_e*, and make *Ku0_e* far too less than *Ku1_e*, thus curve *u_power_i* is very sharp, only when *e_i* is very little, then *u_power_i* would approach to 1, otherwise, *u_power_i* is always very little, thus, those *u_i* that correspond to biggish *e_i* are filtered off, however, those *u_i* that correspond to minor *e_i* are selected. So, algorithm of formula (9) change into the algorithm of formula (10) with selection, and better efficiency would be get.

Of course, what mentioned above are only close to, not equal to so-called "setting value" or "right value", but for practical project in engineering, it is enough to incarnate the idea of "closing to"; If *k_u*=0, then output *u* of CISAPID would not relate to u_0. If "setting value input" of system is a constant, then *k_u* could be greater than 0.5, or even near to 1. In fact, The Analytic Functions of formula (9) and (10) incarnate the ideal of "stabilizing by force" and "Sampling and Statistical Learning", and computer is always in a state of self-studying and self-perfecting, the more it has learned, the more it become "clever" [8]. Statistical Learning has solid theoretical basis, and is drove fully by objective data, so, playing an important role in CI. Basal model, strategy and algorithm related to the design of Statistical Learning perhaps is a direction needed our efforts in future.

Practical running of the algorithms mentioned above shows: With u_0, CISAPID can shorten control time notably if input is step signal, can come into stable-state and close to "setting value" or "right value" more quickly, and can keep stable-state for a comparatively longer time. But, if system is simple and is very stable, then, it is not necessary to make use of this algorithm;If system is tracking system and the requirement for rapidity is high, it is also not appropriate to make use of this algorithm.

4. SIMULATION[6]

Typical object G(S)= $\dfrac{e^{-5s}}{(60s+1)(50s+1)}$ is taken as controlled object to carry out simulation. As to such second-order object with great inertia, great hysteresis in practical project, what is the most important may be stability and rapidity but not accuracy or control time (is allowed to correspond to 5% (or >5%) error range). There are four criterion for performance comparison during simulation: 1.Integral Squared Error (ISE), (let J= $\int e^2 dt$); 2. Rise time (defined as the time needed that system rise from zero to 90% steady-state value); 3.Average value of $|e|$; 4.Overshoot;

ISE and Rise time are main criterions. Arguments of General PID (perhaps had been optimized in reference [6]) are as follows: Kp=5, Ki=0.025, Kd=90; Arguments of CISAPID (only satisfactory but not optimized) are as follows: Kp0_e=4.999, Kp1_e=217, W_Kp_e=100, p_s=0.085, Kp_s=0.0002119, Ki0_e=0.015, Ki1_e=0.1, W_Ki_e=100, Kd0_e=90, Kd1_e=100, W_Kd_e=100, d_s=0.2, Kd_s=2

4.1 Performance comparison

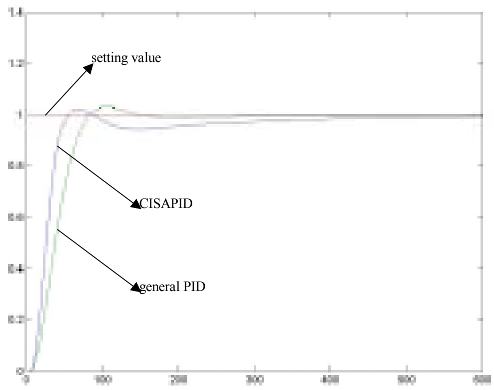

Fig.1 Comparison between CISAPID and general PID for ideal case

When the case is ideal, and there are no disturbance and non-linear parts, the performance of general PID is very well, but it is still inferior to that of CISAPID. General PID: ISE is 29.29, average value of |e| is 0.07128, rise time is 66 seconds, overshoot is 3.014%; CISAPID: ISE is 21.15, enhancement is 27.79%, average value of |e| is 0.06706, enhancement is 5.92%, rise time is 42 seconds, enhancement is 36.364%, overshoot is 2.319%, enhancement is 23.06%;

If controlled object changed greatly, for instantce, hysteresis changed to 8 seconds, pole points changed to 1/110 and 1/100, simulating curves are as follows:

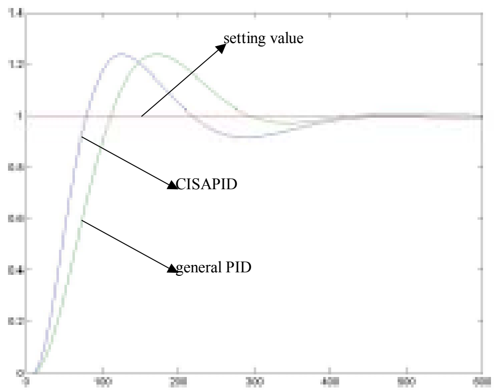

Fig.2 Comparison between CISAPID and general PID if controlled object had changed

Compare with Fig.1: general PID: ISE is 55.38, 55.38-29.29=26.09, average value of |e| is 0.1581, 0.1581-0.07128=0.08682, rise time is 100 seconds, 100-66=34, overshoot is 23.91%, 23.91%-3.014%=20.896%; CISAPID: ISE is 41.56, 41.56-21.15=20.41<26.09, average value of |e| is 0.1316, 0.1316-0.06706=0.06454<0.08682, rise time is 71 seconds, 71-42=29<34, overshoot is 23.88%, 23.88%-2.319%=21.561%>20.896%;

From above we know that the performance of general PID and CISAPID are both worsen when controlled object changed, but the worse of CISAPID are less than those of general PID except overshoot.

4.2 Comparison for Anti-disturbance

We could suppose that disturbance is a sinusoidal input of which the amplitude is 0.2 and the frequency is 0.0314 radian / second (cycle is 200 seconds), then, simulating curves are as follows:

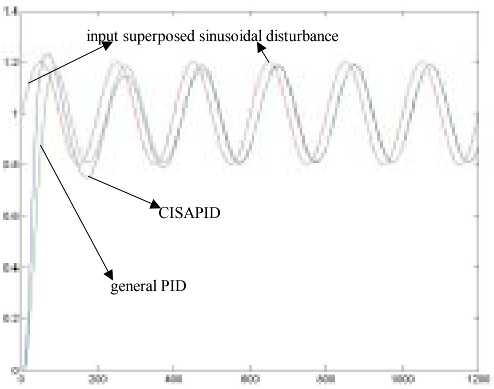

Fig.3 Comparison between CISAPID and general PID if disturbance is sinusoidal

It can be known from the Fig.: general PID: ISE is 46.38, average value of |e| is 0.1207, rise time is 65 seconds, overshoot is 19.38%; CISAPID: ISE is 34.94, enhancement is 24.67%, average value of |e| is 0.1065, enhancement is 11.765%, rise time is 44 seconds, enhancement is 32.31%, overshoot is 24.28%, enhancement is - 25.28%; When input is changing, the requirement for rapidity is main to make system follow the change of input quickly enough, it is allowed that overshoot increased a bit. It also can be known from the Fig.: the output of CISAPID and general PID are both lag behind input, but, after system become stable, the lag of CISAPID is less about 1 second than that of general PID.

If load (y_d) decreased 0.2 suddenly at t=400, and increased y_d=0.2 suddenly at t=700, and setting value of input increased 0.2 suddenly at t=1000, then, simulating curves are as follows:

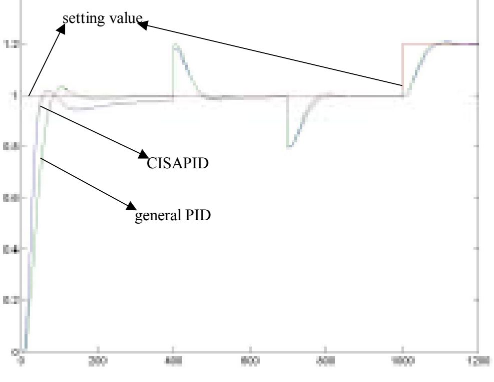

Fig.4 Comparison between CISAPID and general PID if load changed suddenly and disturbance is a step input

It can be known from the Fig.: general PID: ISE is 32.74, average value of |e| is 0.05612, rise time is 66 seconds, overshoot is 3.014%; CISAPID: ISE is 24.63, enhancement is 24.77%, average value of |e| is 0.05451, enhancement is 2.87%, rise time is 42 seconds, enhancement is 36.364%, overshoot is 2.319%, enhancement is 23.06%.

4.3 Comparison for non-minimum phase

Add a non-minimum phase part on the base of previous controlled object, then, transfer function is G(S)= $\dfrac{e^{-5s}(1-s)}{(60s+1)(50s+1)}$ instead, simulating curves are as follows:

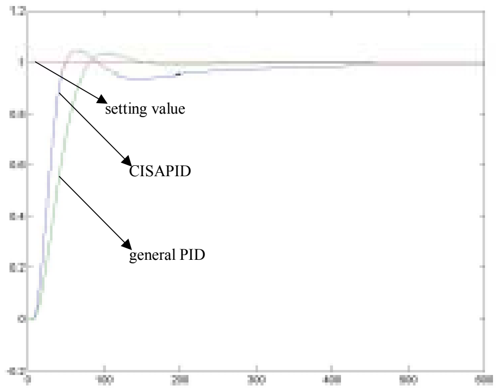

Fig.5 Comparison between CISAPID and general PID if there is a non-minimum phase part

It can be known from the Fig.: general PID: ISE is 29.91, average value of |e| is 0.07195, rise time is 66 seconds, overshoot is 3.48%; CISAPID: ISE is 21.95, enhancement is 26.613%, average value of |e| is 0.06958, enhancement is 3.294%, rise time is 41 seconds, enhancement is 37.88%, overshoot is 4.735%, enhancement is - 36.06%, this is because rise time reduced greatly, rapidity improved greatly, so, overshoot increased a bit more;On the whole, if there is a non-minimum phase part, performance of CISAPID is still better than that of general PID greatly (main criterion ISE improved 21.95%,rapidity(rise time) improved 37.88%).

5. CONCLUSION

The constitution, principle and qualitative arguments tuning experience of an ameliorative PID controller-Computational Intelligence Self-Adaptive PID Controller (CISAPID) has analysed in detail, and the ability for anti-disturbance, robustness, adaptability and the performance for non-minimum phase system of CISAPID are also discussed. Contrastive simulation between CISAPID and general PID showed that the efficiency of CISAPID is better than that of general PID, and practical application in engineering (High Temperature tunnel kiln in TEGAOTE special kiln Corporation, Sansui, GuangDong Province) also showed that the performance of CISAPID is practical, feasible, satisfactory, reliable but simple, better than that of general PID

5.1 Innovative ideas of CISAPID

Here, we should first thank QingChang Zhong, JianYing Xie, Hui Li, please refer to reference [6] for specific details, they bring gaussian function into PID and put forward a new PID controller- Variable Arguments PID (VAPID). But, the constitution of proportional gain function and differential gain function of VAPID are all have something to be improved, and they were probably not aware of : If this idea improved appropriately, it is in fact a new idea for CI with Analytic Functions and Logics at which computer is good. Then, this new idea only process bottom data, and possess the part for pattern recognition (by means of Analytic Functions and Logics), and don't make use of knowledge in the sense of AI, but can incarnate knowledge, rules and experience of experts or operators and can perform Fuzzy Logic Control and Expert Control to a certain extent, thus, this system is a CI system in fact. So-called innovative ideas in this thesis are listed simply as follows, please refer to my master's degree for more details, and we urgently welcome precious critical advice.

1) A new idea - incarnate Computational Intelligence with analytic functions and Logics
2) Particular structure of CISAPID
3) Proportional action of CISAPID
4) Differential action of CISAPID: (1) Based on e; (2) Based on de; (3) Based on d^2e
5) Dynamical Weighting Average Algorithm with selection of CISAPID

5.2 The problems to be improved for CISAPID

There are many arguments of CISAPID, it is a little inconvenient to tune so many arguments, and we have not yet find perfect tuning method in theory, all these are to be improved in future.

As to so many arguments that to be tuned and optimized, resort to Neural Network, Genetic and Evolutive Algorithms maybe good ideas.

Sampling and Statistical Learning, and the basal model, strategy and algorithm related to the design of Statistical Learning perhaps is a direction needed our efforts in future.

At last, it is needed to point out: The reasons why we attempt to incarnate but not to realize or imitate are mainly as follows: 1) It is easy to incarnate, but it is difficult to realize or imitate. 2) Algorithms are comparatively concise, feasible, ingenious, but they are enough to achieve given target for practical application in engineering.

But all these are still only the execution but not the creation of intelligence of human being, in the long run, research for CI should probably aim at the purpose of creating intelligence.

REFERENCES

[1] A A Rovira, P W Murrill, C L Smith. Tuning Controllers for Set-Point Change Instruments and Control Systems, 1969,42(12)

[2] Cohen G N, Coon G A. Theoretical Considerations of Retarded Control. Trans. ASME, 1953,75:287

[3] FuChun Sun, ZengQi Sun Technique on Computational Intelligence (in chinese)

[4] Lopez A M et al. Controller Tuning Relationships Based on Integral Performance Criteria. Instrumentation Technology.1967, 14:57

[5] MingZhong Yi Arguments Self-learning PID Control system with Prediction of BP-NN based on PCC Transactions of 14TH Conference of Control and Decision 2002.5: 484～492(in chinese)

[6] QingChang Zhong,JianYing Xie,Hui Li PID controller with Variable Arguments (in chinese) 《Information and Control》Vol.28 No.4 1999

[7] Unscramble Intelligence Computation in 21th Century the third International Conference on Computation in 21 Century Microsoft Research China (in chinese)

[8] Unscramble Intelligence Computation of Microsoft the third International Conference on Computation in 21 Century(in chinese)

[9] Ziegler J G,Nichols N B. Optimum Settings for Automatic Controllers.Trans. ASME, 1942,64:759

[10]森, 重政, PID オートチェニクコニトロテの動向, 計測と制御 1990, 29(8): 29～34