**Lecture#11 part2: Type of controllers**

**5- Robust PID tuning method:**

The design procedure consists of three steps:

1. Select the of the closed loop system by specifying the setting time(*ts*) .
2. Determine the three coefficients(Kp , Ki , Kd) using the appropriate optimum equation (Table(\*)) and the of step 1 to obtain *G*c(s) so that the close loop system transfer function , *T*(s), does not have any zero, as required by equation (\*):

(\*)

**Table (\*):** The Controller Equation Robust

|  |
| --- |
| **The optimum coefficients of T(s) based on the ITAE criterion for step input** |
|  |
|  |
|  |
|  |

Ex1: Consider a temperature controller with a control system as show in this equation

G(s) = 

the steady state error is 50%, and setting time (2% criterion) is  for a step input and a setting time of less than 0.5 sec., so the  using PID controller we have :



Therefore, the closed loop transfer function is:



The optimum coefficients of the characteristic equation For ITAE are:

After compare these two equation, we get:





-----------------------------------------------

**6-Direct synthesis for tuning PID controller:**

This is a model based tuning technique. It uses an identified process model in conjunction with a user specified closed loop response characteristic. An advantage of this approach is that it provides insight into the role of the 'model' in control system design. A disadvantage of the approach is that a PID controller may not be realised unless an appropriate model form is used to synthesise the control law.

Let the symbol Gp represent the process dynamics and Gc the controller dynamics. If all other dynamic elements within the loop are ignored then the following closed loop transfer function can be derived,



this can be re-arranged to give an expression for the feedback control law as,



The closed loop response characteristic,  must be specified. A simple desired specification is, , λ is a user specified closed loop time constant.

Substituting this equation into equation of Gc and re-arranging gives,



where it has been assumed that the plant transfer function is,



ie. first order, no dead-time.

Based on this process description, the ideal form of a PI controller results,

where,



**What do you do if you want derivative action?** The first order model results in a control law that is of the PI type. If you wish to synthesis a PID controller,there are two options

- choose TD = Ti/4

-Model the process using a 2nd order transfer function.

**Systems with time delay**

IF the plant contained time delay (i.e. a first order plus dead-time transfer function) which has the following equation: 

With , then the controller PI equation becomes



Where

 and Td=Ti/4

--------------------------------------------------------

**Type of PID configurations:** there are three type for connection the PID controller

**1) Ideal PID:**  The mathematical representation of this algorithm is:



One disadvantage of this ideal configuration is that a sudden change in input (and hence e) will cause the derivative term to become very large and thus provide a “derivative kick” to the final control element -this is undesirable. An alternative implementation are:

i)PI\_D controller



The derivative mode acts on the measurement and not the error. After a change in input the output will move slowly avoiding "derivative kick" after input changes. This is therefore a standard feature of most commercial controllers.

ii) I\_PD controller



iii) In an actual PID controller, instead of the pure derivative term *T*d*s,* we employ



where the value of  is somewhere around O.l.Therefore, when the reference input is a step function, the manipulated variable *u(t)* will not involve an impulse function.

***2)Series (interacting) PID*:**  The mathematical representation of this algorithm is:



As with the ideal implementation the series mode can include either derivative on the error or derivative on the measurement. In which case, the mathematical representation is,

 *where*  e(s) *= r*(s) *-* TDsy(s)

**3)Parallel PID:** The mathematical description is,



The proportional gain only acts on the error, whereas with the ideal algorithm it acts on the integral and derivative modes as well.

**Revision Exercises**

1. Draw the block diagram representation of the ideal, series (interacting) and parallel PID control laws.

2. Write down the 'time-domain' mathematical representation of the ideal (without derivative kick), series (interacting) and parallel PID control laws.

