MECHANICAL ENGINEERING FACULTY

PROJECT II

Comparison of modified of PID controllers

Submitted by : Ing. Stanislav Vrána, PhD.

Student : Caner Emrecan

CONTENTS

1.ASSIGMENT

2.What is PIDplus and its comparasion with PID?

3.Comparasion and convertantion Discrete PID controller to Continuous PID Controller

4. 2-DOF PID Controller and comparison with PID controller

1.ASSIGMENT

Comparison of modified of PID controllers

The PID controller is the most used type of controller. To expand its capabilities, extended modifications of this controller were created, e.g. PIDplus , FPID, which is based on the calculation of a non-integer integral and derivative, fuzzy PID controller, and others. Search for existing extensions of the standard PID controller and compare their capabilities and limitations using the Matlab/Simulink program. Do not focus only to the mentioned modifications.

2. What is PIDplus and its comparasion with PID?

A.PID Controller

Proportional Integral Derivative (PID) Controller The PID controller is operated based on summation of three term; P-term (which operate based proportional on error), I-term (which operate based on proportional to integral on error), and D-term (which operate based on proportional to derivative on error). For this paper, all PID structures are designed in non-interacting form based on Equation 1 and it implementation in Simulink is illustrate in Equation. 1.

$$G(s) = K_p + \frac{K_i}{s} + K_d s \tag{1}$$

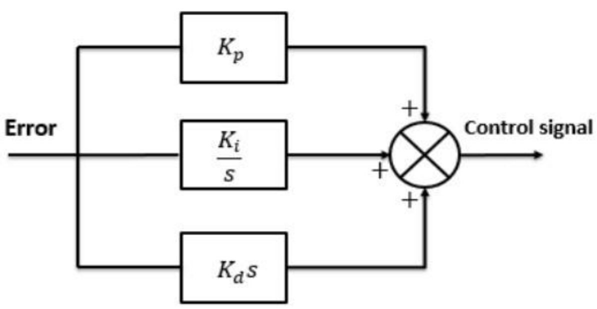

Fig.1. PID controller [1]

B. PIDplus

In PIDPlus controller design, the integral (I) and derivative (D) control actions of standard PID controller are modified to deal with delayed process measurement updates and communication loss.Both actions are updated only when controller receives new process measurements. During the communication loss intervals, these actions are frozen to prevent integral windup and derivative kick problems [2].

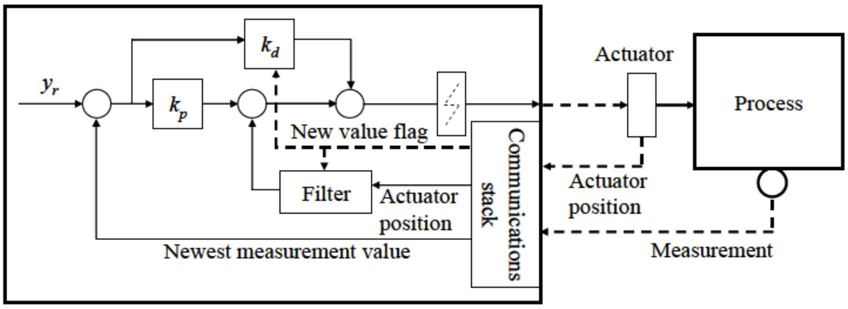

Fig.2. PIDplus computation scheme [3]

The filter equation:

$$F(k) = F(k\text{-}1) + (O(k\text{-}1) - F(k\text{-}1))\,(1 - e^{-\Delta T/Ti})$$

(2)

F(k): Filter output

F(k-1) = Last filter output

O(k-1) = the last controller output

ΔT = time between recievedpackages

Ti = P/I

The derivative part of the proposed algorithm is simply an event-based version of the regular D-term, and it is given by

$$D(k) .= k_d\,(e(k) - e(k\text{-}1)/\,\Delta T)$$

(3)

e(k) :current error

e(k-1) : last error

In practical implementations a measurement filter is also required as in (3). In this paper, we propose the following first-order filter to be used with the PIDPLUS algorithm:

$$e_f(k) = \frac{T_f}{\Delta T + T_f} e_f(k-1) + \frac{\Delta T}{\Delta T + T_f} \left(\underbrace{c y_r(k) - y(k)}_{e(k)} \right) \quad (4)$$

Here *ef* is the filtered error signal, which is then used in the derivative term of the controller instead of the pure error signal *e*. Note the dynamics of the filter: for long outages (large ΔT) the filter weights the up-to-date raw data more than the outdated filtered value (previous *ef*). If there are no outages, the speed and efficiency of the filter are determined by Tf / h ratio. Here h is the sample time, which is a typical value of ΔT [4]

C. Control Simulation

The simulation of PIDPlus control is built in the Simulink system of MATLAB, as shown in Figure 3. Among them, the setting value is unit step signal; the controlled object is the representative first order $G(s) = \frac{2}{3s+1}$ the controllers are developed by S-Function module, in which customized PID controller is realized by traditional positional PID algorithm and PIDplus controller is realized by improved PID algorithm. The parameters of PID are Kp = 0.8,Ki = 0.16,Kd = 0.32, which are acquired by PID Tuner tool of MATLAB by default.

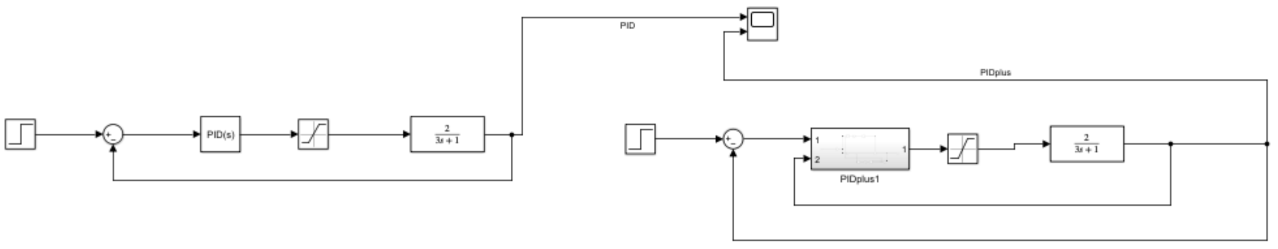

Figure 3.Matlab/Simulink Model of PIDplus and PID controller

Fig. 4 is a comparison of PID and PIDPlus control without transmission delay. Theoretically, the working effect of PID and PIDPlus should be exactly the same, but the experimental results show that even for normal case, PIDPlus will make the control process smoother. And of course, the corresponding response time becomes longer at the same time. The reason may be that the output of the controller changes too little in the adjacent sampling period, or the controlled variable itself has a lag characteristic, and the controlled variable does not change between the two sampling periods, which results in the different curves. After setting the sampling time to 300s and the time delay to 0, the simulation results show that the PID and the PIDPlus work exactly the same.

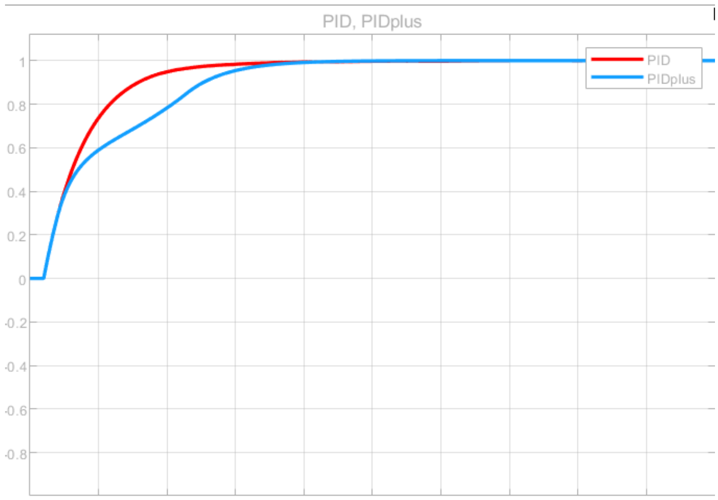

Figure 4.Contrast of PID and PIDplus

Fig. 5 is a comparison between PID and PIDPlus control with variable period delay. It can be seen that in the case of large variable period delay, PIDPlus can significantly make the transition process smoother. The effect of PIDPlus control is much better than that of PID control.

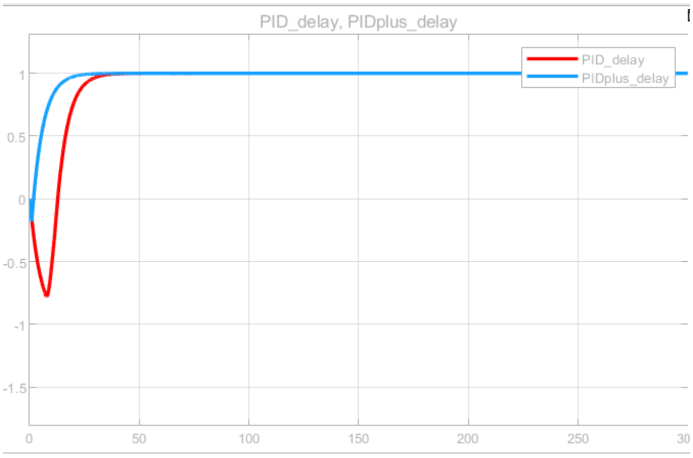

Figure 5.Contrast of PID and PIDplus with transport delay

The simulation results show that the PIDplus controller achieves faster settling time and lower overshoot than the PID controller.

Specifically, the settling time of the PIDplus controller is approximately less than 25 seconds, while the settling time of the PID controller is approximately 40 seconds.

3. Comparasion and convertantion Discrete PID controller to Continuous PID Controller

A discrete PID controller will read the error ,calculate and output the control input at a given time interval,at the sample period T.The sample time should be less than shortest time constant in the system.

Unlike simple control algorithm,the PID controller is capable of manipulating the process inputs based on history and rate of change of the signal.The gives a more accurate and stable control method.

- Kp = 6
- Ki = 12
- Kd = 1

```
Kp = 6;
Ki = 12;
Kd = 1;
N = 100;

% Define the plant and the controller in the s-domain.
TFPlant = 0.09/(0.09*s^3 + 0.18*s^2 + s + 1.004);
TFPIDFilt = Kp + Ki/s + Kd*N/(1 + N/s);

% Obtain the numerator and the denominator of the transfer functions in the
% s-domain.
[NPIDCont, DPIDCont] = numden(collect(TFPIDFilt));
NPIDCont = sym2poly(NPIDCont);
DPIDCont = sym2poly(DPIDCont);

% Convert to z-domain using Tustin substitution (referred to as Trapezoidal
% method in Simulink block PID(s)).
TFPlant = collect(subs(TFPlant, s, (2/T)*(z - 1)/(z + 1)));
TFPIDFilt = collect(subs(TFPIDFilt, s, (2/T)*(z - 1)/(z + 1)));

% Define time step for discrete simulation.
Tval = 0.1;

% Perform substitution for the time step T.
TFPlant = subs(TFPlant, T, Tval);
TFPIDFilt = subs(TFPIDFilt, T, Tval);

% Decompose into the numerator and denominator.
[NPlant, DPlant] = numden(TFPlant);
[NPIDFilt, DPIDFilt] = numden(TFPIDFilt);

% Obtain the polynomial coefficients associated with the numerator and
% denominator.
NPlantCoeffs = sym2poly(NPlant);
DPlantCoeffs = sym2poly(DPlant);
NPIDFiltCoeffs = sym2poly(NPIDFilt);
DPIDFiltCoeffs = sym2poly(DPIDFilt);
```

Figure 6. The script for the definition of the parameters for the discrete-time simulation.

For the discrete time simulation, it is important to select the fixed time step solver and set the time step to the value that is equivalent to the value that was used for the conversion of the plant model from s-domain to z-domain. As you can see below, I used the z-domain version of the continuous time PID controller block for the discrete time simulation. Given the calculated plant parameters for the discrete-time simulation, the result of the simulation is very close to the result of the simulation of the continuous time system.

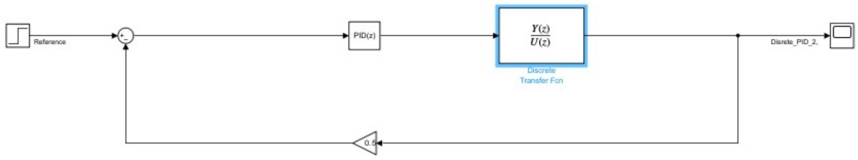

Figure 7.Simulink Representation of discrete PID controller

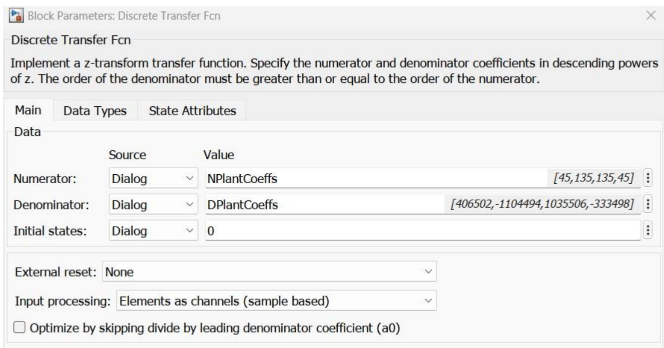

Figure 8. Discrete time plant model parameters

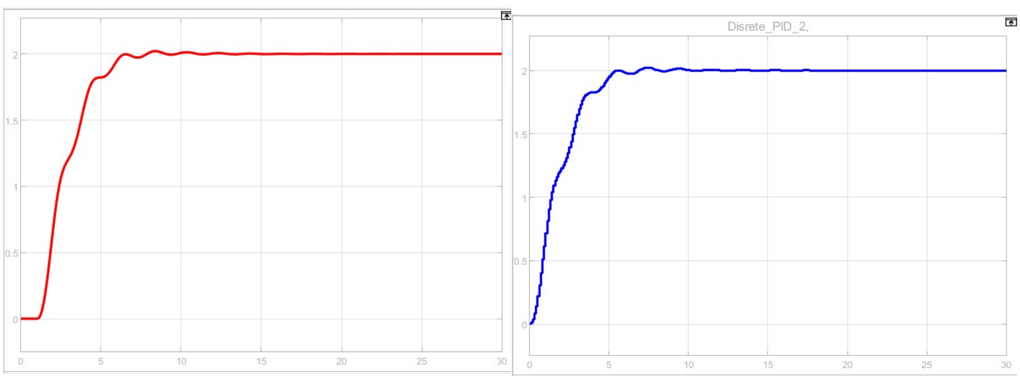

Figure.9 The result of the continuous and dicrete PID time simulation

Time Domain Parameter	Conventional PID	Discrete PID
Rise Time(s)	6.7	5.2
Settling Time(s)	18.2	16.1
Peak Overshoot(%)	16.55	9.03
Peak Time(s)	8.5	7.5

From the results, Discrete PID controller provides better time domain parameter enhancing performance and system efficiency. It is noted that Discrete PID controller has a reduced rise time of 5.2 sec against PID 6.7 sec. And it is found that Discrete PID controller has reduced Peak over shoot of 9.0383% over PID controller with 16.5567%. Hence it is clearly noticed that settling time, rise time and peak time are greatly reduced in the case of Discrete PID controller method compared to conventional PID controller is better than conventional PID controller.

4. 2-DOF PID Controller and comparison with PID controller

A. What is Two Degrees Freedom of PID Controller?

- The 2DOF PID controller, unlike the traditional PID controller, has two separate inputs: one for monitoring the setpoint change (reference change) and the other for monitoring the measurement error.

- This allows the controller to perform two independent recovery processes: one to ensure fast setpoint tracking and the other to ensure measurement error is close to zero.

- The 2DOF PID controller can provide greater flexibility in terms of performance and stability, because it is possible to tune two separate inputs.

- Traditional PID controller is generally used in simple applications and linear systems.

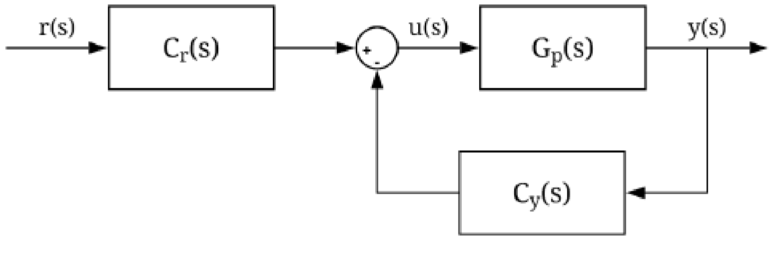

Figure 10.Block Diagram of the 2-DOF Controller [5]

- A 2DOF PID controller may be more effective, especially in situations that require fast setpoint tracking or where the measurement error must approach zero quickly. 2DOF PID controller is often used in complex systems and to provide greater control to meet specific performance requirements.

As a result, a traditional PID controller or a 2DOF PID controller may be preferred depending on your application requirements and control system performance goals. A 2DOF PID controller can provide greater flexibility and control in certain situations, but these situations will vary depending on your specific system requirements.[6]

B.Control,Comprasion and Simulation

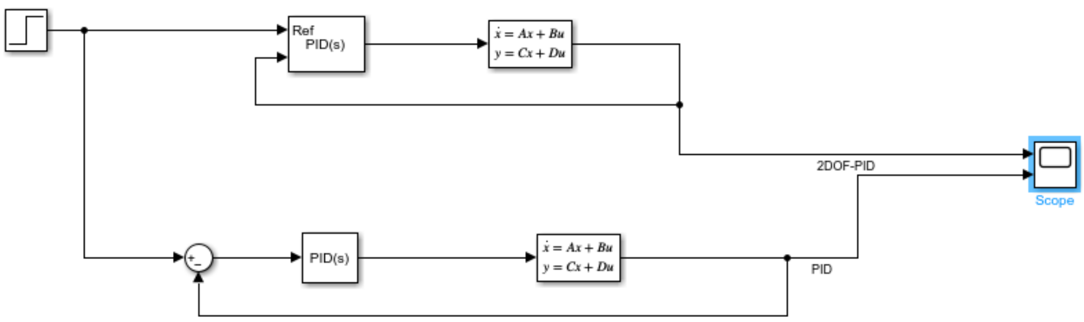

Figure 11. Matlab/Simulink Model of 2-DOF PID and PID controllers

Controller	Kp	Ki	Kd	α	β	γ
PID	0.015	0.003	0.0035	-	-	-
2-DOF PID	0.015	0.003	0.0035	50	0.8769	0

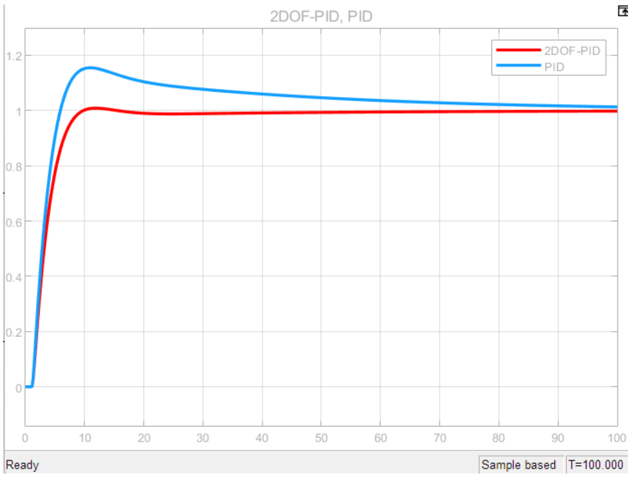

Figure 12. The result of the continuous and dicrete PID time simulation

The control system uses a PID controller, while the second employs a 2–DOF PID controller. In general, both controllers achieved similar performance in experimental test,.However, the step response of the 2–DOF PID controller showed significantly less overshoot and steady state error and similar settling time. The main reason for this difference is that the PID controller was not able to compensate the oscillations in the steady state produced .The amplitude of the oscillations in the PID control system tended to increase slowly with time.

References

[1] M. I. P. Azahar, A. Irawan and R. M. Taufika, "Fuzzy Self-Adaptive PID for Pneumatic Piston Rod Motion Control," *2019 IEEE 10th Control and System Graduate Research Colloquium (ICSGRC)*, Shah Alam, Malaysia, 2019, pp. 82-87, doi: 10.1109/ICSGRC.2019.8837064

[2] T. Blevins, M. Nixon, M. Zielinski, Using wireless measurements in control applications, International Society of Automation (ISA) (2103).

[3][4] O. Kaltiokallio, L. M. Eriksson and M. Bocca, "On the performance of the PIDPLUS controller in wireless control systems," 18th Mediterranean Conference on Control and Automation, MED'10, Marrakech, Morocco, 2010, pp. 707-714, doi: 10.1109/MED.2010.5547788.

[5] D. Shatwell, F. Salazar and A. Rojas–Moreno, "2–DOF PID Control of the Angular Position of an Industrial Plant Emulator," *2020 IEEE XXVII International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, Lima, Peru, 2020, pp. 1-4, doi: 10.1109/INTERCON50315.2020.9220262.

[6] Adar, N., i R. Kozan. „Comparison between Real Time PID and 2-DOF PID Controller for 6-DOF Robot Arm". *Acta Physica Polonica A*, t. 130, nr 1, Polska Akademia Nauk. Instytut Fizyki PAN, 2016, s. 269–71.