Arduino PID Practice Problems

In Figure 1 a schematic of a system with a PID controller is shown.

1. What does variable y represent?

   (a) setpoint value , (b) measured process value, (c) the difference or error, (d) process input (e) none of the above

2. What does variable u represent?

   (a) setpoint value , (b) measured process value, (c) the difference or error, (d) process input (e) none of the above

3. What does the input u try to do?

   <u>This input will try to adjust the measured process value back to the desired setpoint.</u>

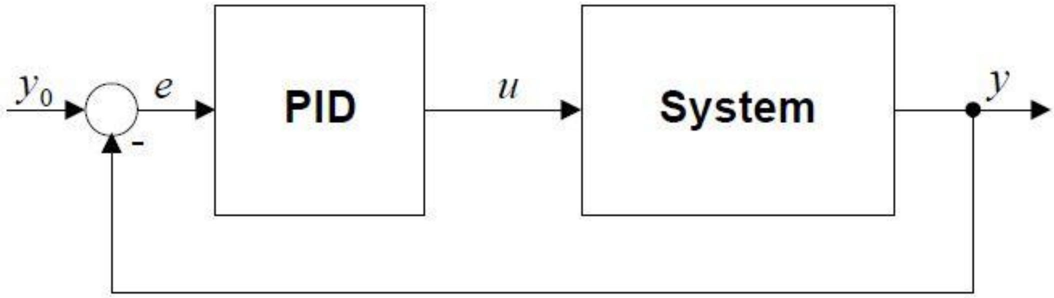

Figure 1 Closed Loop System with PID controller

4. A simple *proportional* control algorithm differs from a PID controller by not looking at

   (a) The present (b) The past (c) The future (d) answers b and c (e) all of the above

5. For most control applications, a simple *proportional* control algorithm after the system has stabilized will always have a…

   (a) dynamic error component (b) certain amount of instability (c) error frequency (d) stationary error (e) none of the above

6. A PID controller generates the process value (u) by looking at…

   (a) The present (b) The past (c) The future (d) answers b and c (e) all of the above

7. The PID controller can look into the future by …

   (a) looking at the rate of change of the signal (b) looking at the history of the signal (c) using a crystal oscillator (d) integrating the signal over time (e) answers b and d

8. The signal input to the PID controller is generated by

(a) subtracting the measured process value from a desired reference value (setpoint) (b) answer a is not possible because these two values have different units (c) answer b doesn't know what he is talking about (d) answers a and c (e) subtracting the process output by the process input

9. Figure 2 shows the PID controller schematics. What does Kp represent?

   (a) the time constant of the proportional term, (b) the time constant of the integral term, (c) the time constant of the derivative term, (d) the constant of proportionality (e) none of the above

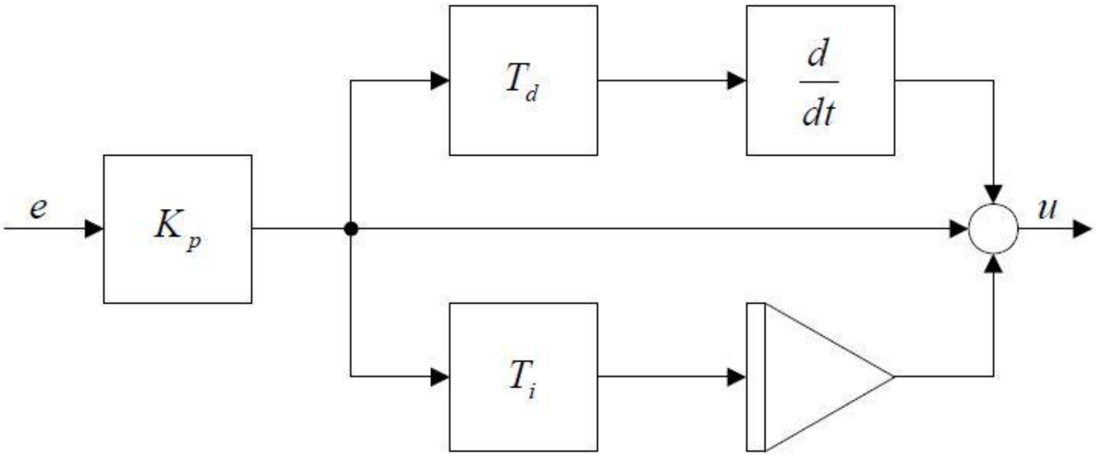

Figure 2 PID controller schematic

10. Which of the following applications would not benefit from a PID controller

    (a) motor control, (b) control of temperature, (c) speed, (d) flow rate, (e) none of the above,

11. In Figure 3 the red line represents a reference (setpoint) input to a closed system. The measured process value is shown in blue. What type of controller is used in this application?

    (a) p (b) pi (c) pid (d) pd (e) state space model

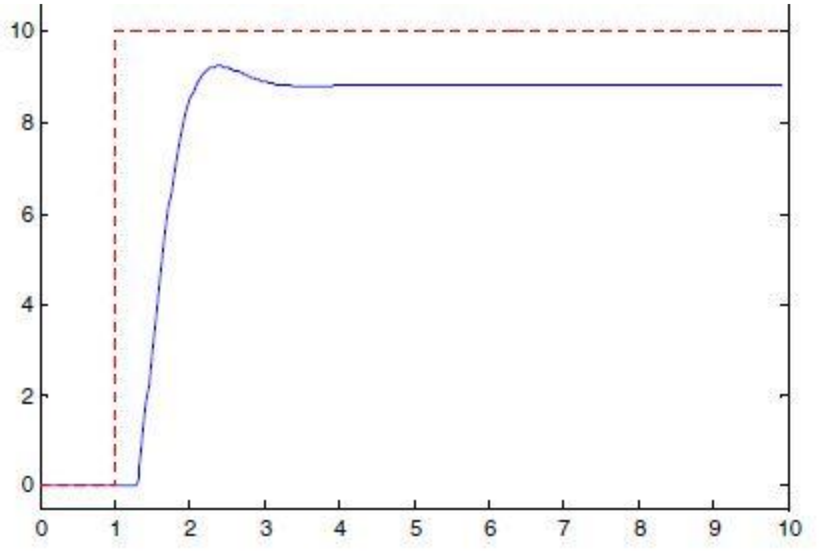

Figure 3 controller schematic

12. The integral term I in a PID controller gives…

    (a) a system control input proportional with the error. (b) an addition from the sum of the previous errors to the system control input. (c) an addition from the rate of change in the error to the system control input. (d) I found the answer to this question in the ATMEL article (e) answers b and d

13. What term improves the response of the controller and reduces overshoot?

    (a) proportional (b) integral (c) derivative (d) answers a and b together (e) none of the above

14. System windup adversely effects the calculation of what PID term?

    (a) proportional (b) integral (c) derivative (d) answers a and b together (e) none of the above

15. The PID control software provided as part of the Arduino PIDLibrary differs from both the Bare Bones Coffee Controller BBCC and the AeroQuad in what ways.

    (a) It is more complex, (b) Its inputs are normalized, (c) inputs and output limits can be set, (d) it supports multiple control loops (e) all the above [note: the answer to this question is d]

16. Which of the following includes an “industrial-standard” reset-windup mitigation code?

    (a) Arduino PIDLibrary (b) Bare Bones Coffee Controller (BBCC) (c) AeroQuad (d) none of the above (e) all the above.

17. You can tune a PID controller (1) experimentally, (2) by measuring the step response of the system, and by (3) measuring the stable period of oscillation about a setpoint. Which of these three approaches allows tuning on the fly?

(a) method 1 (b) method 2 (c) method 3 (d) methods 1 and 3 (e) all the methods

PID Control Software Sample Questions

```
double err = 0; // error term
// time constants
// these values are set by tuning the system
double Kp;     // proportional term
double Ti;     // integral term
double Td;     // derivative term

// PID terms
double P;  // proportional
double I;  // integral
double D;  // derivative

err   ______________________________

D     ______________________________
I     ______________________________


double u = Kp * (err + Ti * I - Td * D);

last_input = input;
```

18. Write the C++ expression to calculate the error term

```
Ans: err = ref - input;
```

19. Write the C++ expression to calculate the derivative term

```
Ans: D = input - last_input;  // dT = constant
```

20. Write the C++ expression to calculate the integral term

```
Ans: I += err;
```

21. The process value (output) equation **u** includes the proportional term (p). What variable represents this value?

```
Ans: P = err;
```

22. To prevent reset-windup write a C++ code snip-it to check and see if the output is pegged at a limit and only integrate if it is not.

```
Ans: if (!(lastOutput >= 1 && err>0) && !(lastOutput <= 0 && err<0))
       {
         I += err;
       }
```

23. To prevent the integration term **I** getting huge despite lots of error, the bare bones coffee controller (BBCC) uses a "windup guard" (this happens when the machine is first turned on and it can’t help be cold despite its best efforts). Unlike, the PID library implementation, the BBCC does this by simply limiting the integration term to predefined limits. Write the C++ code for this solution.

```
Ans:  if (I > windupGaurd)
        I = windupGaurd;
      else if (I < -windupGaurd)
        I = -windupGaurd;
```