

CHƯƠNG 6: TỔNG HỢP PHẦN CỨNG VÀ PHẦN MỀM

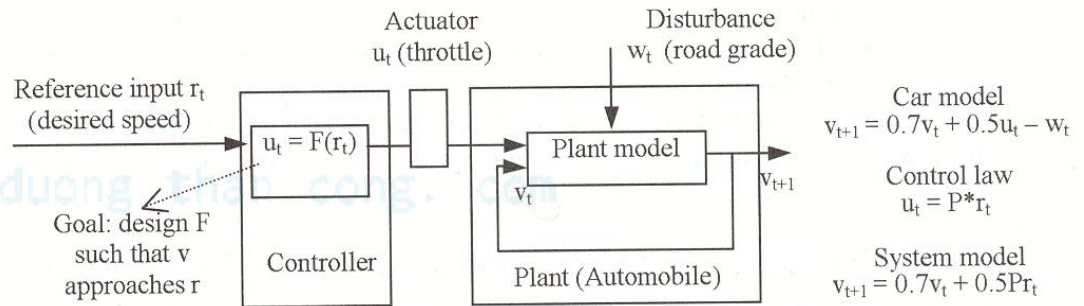
Bài 10: Các hệ thống điều khiển

cuu duong than cong. com

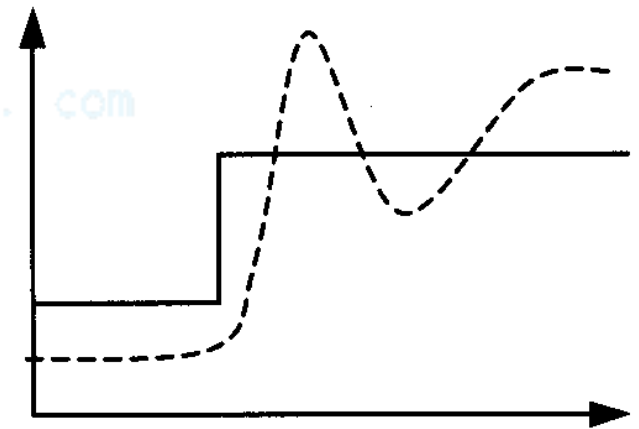
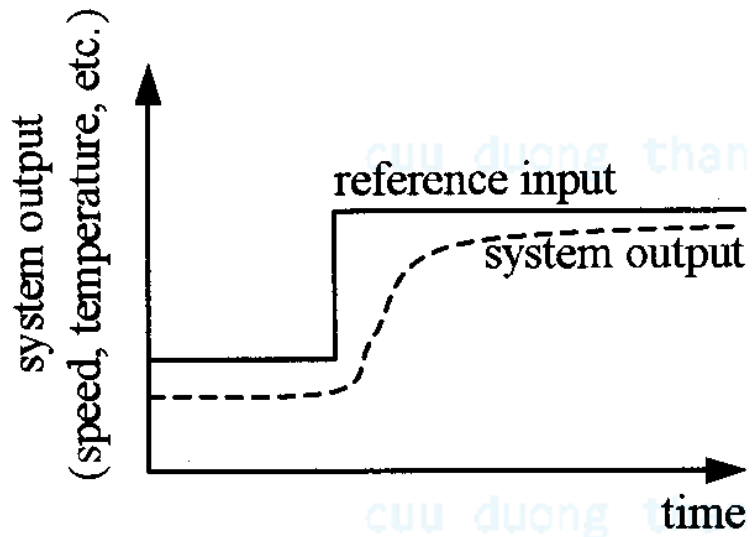
cuu duong than cong. com

Hệ thống điều khiển

- Điều khiển đầu ra của hệ vật lý
 - Bằng cách thiết lập đầu vào của hệ vật lý
- Điều khiển bám
- VD
 - Điều khiển lái
 - Điều khiển nhiệt độ
 - Điều khiển ổ đĩa
 - Điều khiển bay
- Khó, do các nguyên nhân:
 - Nhiều: gió, mặt đường, lốp xe, phanh...
 - Tương tác với con người

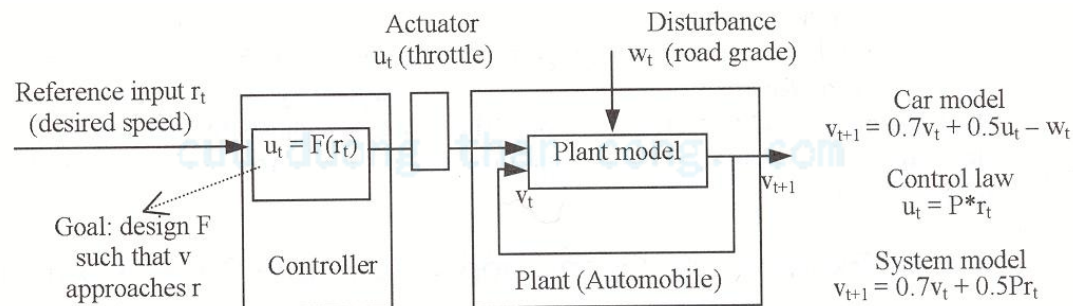


Bám



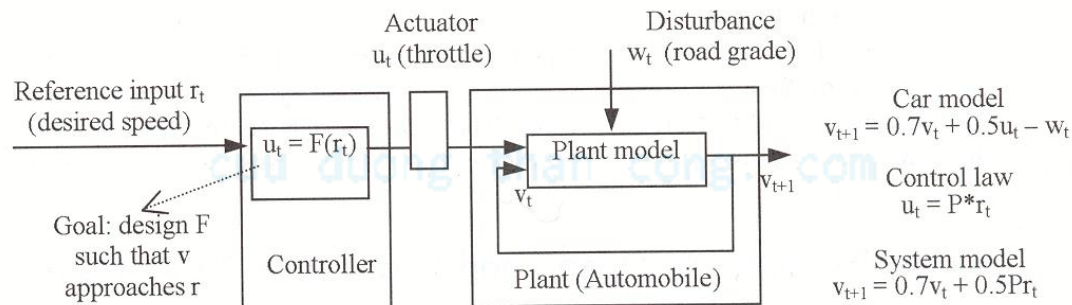
Các hệ thống điều khiển vòng hở

- Đối tượng
 - Các hệ vật lý được điều khiển
 - Ô tô, máy bay, lò nhiệt...
- Cơ cấu chấp hành
 - Thiết bị để điều khiển đối tượng
 - van, động cơ...
- Bộ điều khiển
 - Sản phẩm được thiết kế để điều khiển đối tượng



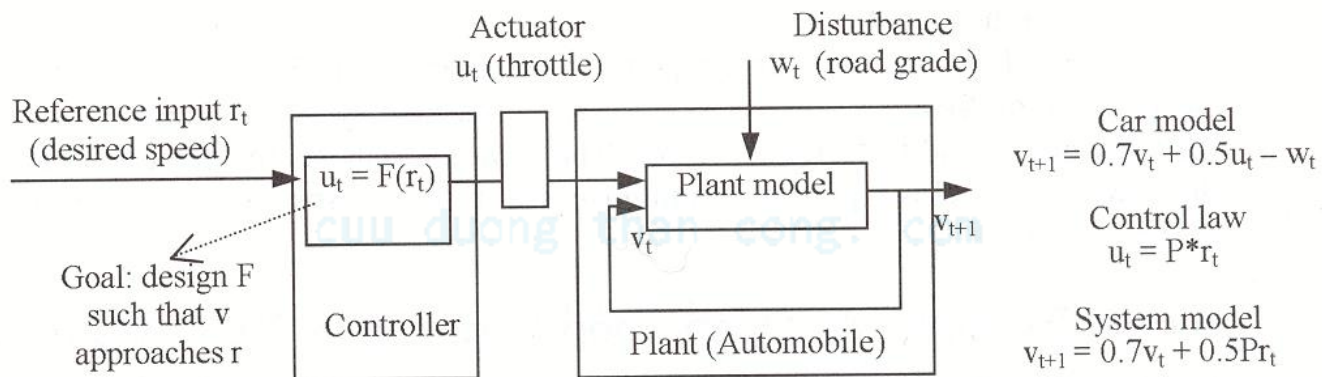
Các hệ điều khiển vòng hở

- Đầu ra
 - Các thông số của đối tượng mà ta quan tâm
 - Tốc độ, vị trí, nhiệt độ,...
- Giá trị tham chiếu
 - Giá trị chúng ta muốn đạt được ở đầu ra
 - Tốc độ yêu cầu, vị trí mong muốn, nhiệt độ mong muốn
- Nhiễu
 - Đầu vào không điều khiển được
 - Gió, lực ngoài



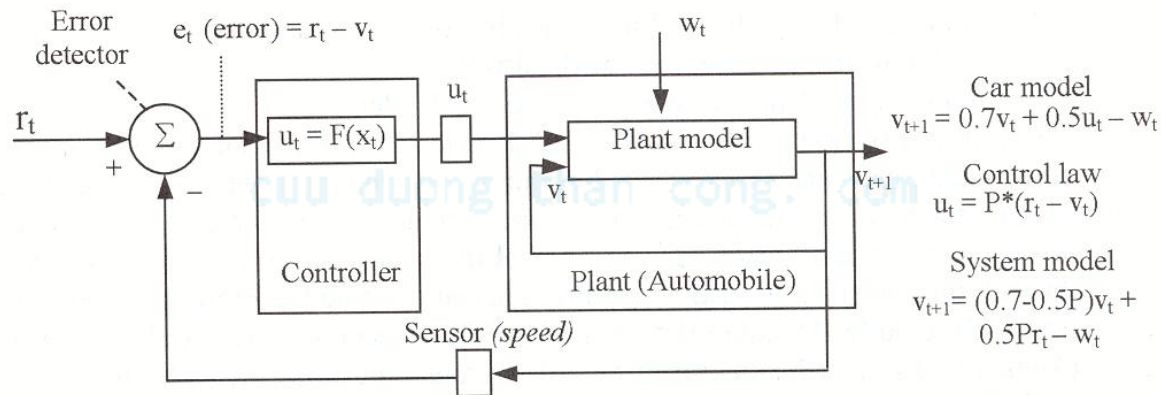
Các đặc tính khác của vòng hở

- Điều khiển Feed-forward
- Có trễ ở đầu ra
- Bộ điều khiển không biết đầu ra có đáp ứng theo yêu cầu
- Đơn giản
- Sử dụng tốt nhất cho các hệ thống có thể dự đoán đầu ra



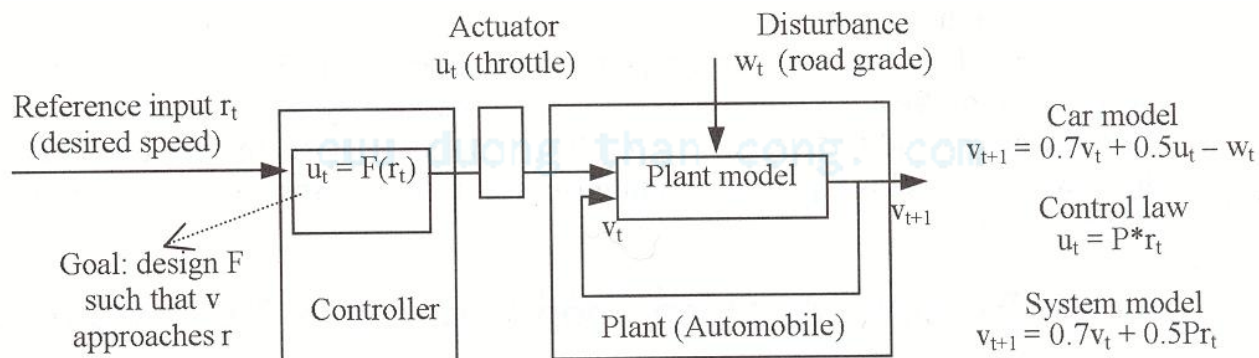
Các hệ điều khiển vòng kín

- Cảm biến
 - Đo đầu ra của đối tượng
- Bộ xác định lỗi
 - Xác định sai số
- Hệ điều khiển vòng kín
- Giảm sai số bám



Thiết kế hệ điều khiển vòng hở

- Xây dựng mô hình đối tượng
- Xây dựng bộ điều khiển
- Phân tích bộ điều khiển
- Xem xét nhiễu
- Xác định chất lượng
- Ví dụ: Hệ điều khiển lái vòng hở



Mô hình đối tượng

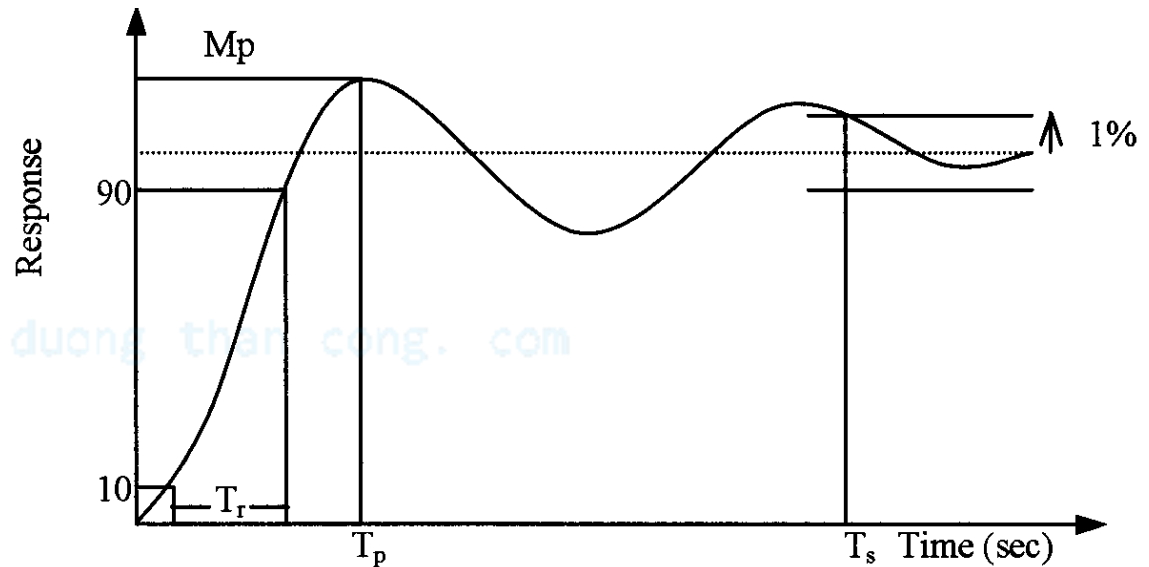
- Có thể không cần thiết
 - Có thể được thực hiện thông qua thực nghiệm hoặc dò
- Tuy nhiên,
 - Có thể giúp quá trình thiết kế đơn giản hơn
 - Có thể sử dụng để tính toán bộ điều khiển
- Ví dụ: điều khiển góc mở từ 0 đến 45 độ
 - Trên mặt phẳng ở tốc độ 50 mph, mở van tới 40 độ
 - Đợi 1 khoảng thời gian nhất định
 - Đo tốc độ, ví dụ 55 mph
 - Dựa vào phương trình sau
 - $v_{t+1} = 0.7 \cdot v_t + 0.5 \cdot u_t$
 - $55 = 0.7 \cdot 50 + 0.5 \cdot 40$
 - Nếu phương trình đúng cho mọi trường hợp
 - Như vậy chúng ta có một mô hình của đối tượng

Hệ điều khiển chung

- Mục tiêu
 - Làm cho đầu ra bám theo giá trị đặt ngay cả khi
 - Nhiễu đo lường
 - Sai số của mô hình
 - Nhiễu ngoài
- Các thông số
 - Tính ổn định
 - Đầu ra nằm trong một giới hạn
 - Chất lượng
 - Đầu ra bám theo giá trị đặt
 - Loại bỏ nhiễu
 - Tính bền vững
 - Khả năng đáp ứng khi mô hình thay đổi

Chất lượng (nói chung)

- Thời gian tăng
 - Thời gian từ 10% tới 90%
- Thời gian đỉnh
- Quá điều chỉnh
 - Phần trăm mà giá trị đỉnh đạt quá giá trị đặt
- Thời gian xác lập
 - Thời gian cần thiết để đạt 1% giá trị cuối

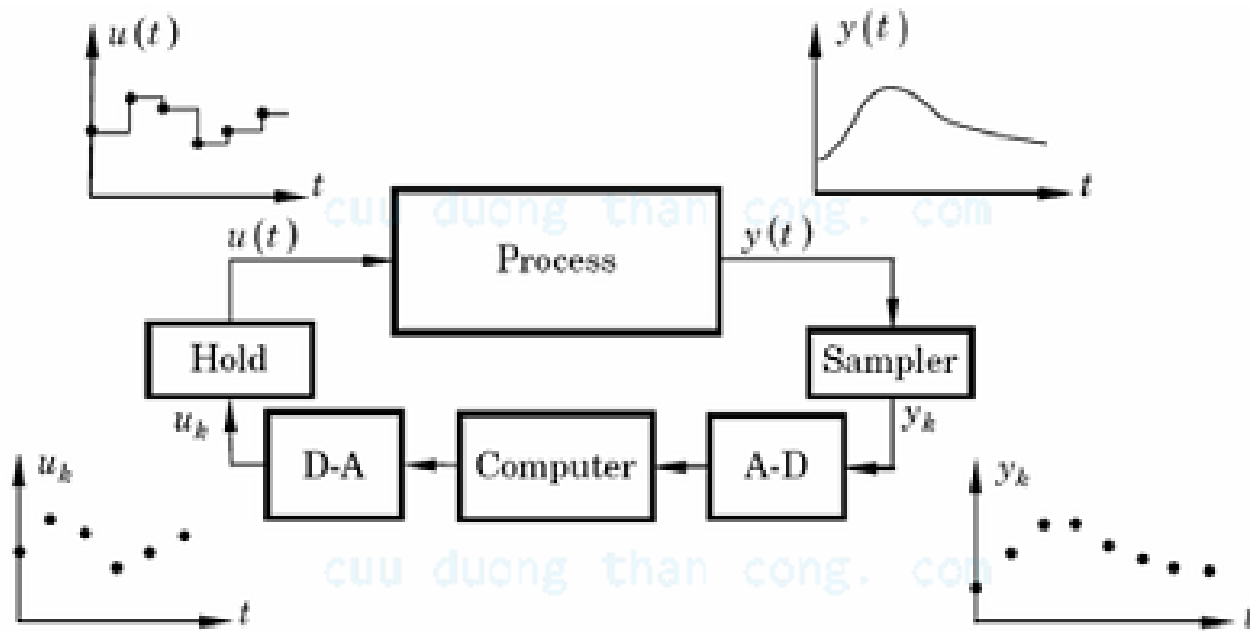


Xây dựng mô hình của đối tượng là khó

- Cần được thực hiện trước
- Đối tượng thường liên tục theo thời gian
 - Không rời rạc
 - VD tốc độ xe thay đổi liên tục so với vị trí góc mở của van
 - Thời gian lấy mẫu phải được lựa chọn cẩn thận
- Đối tượng thường phi tuyến
- Mỗi tương tác giữa mô hình đối tượng và bộ điều khiển
 - Mô hình “đủ tốt” cho việc thiết kế bộ điều khiển

Thực hiện bộ điều khiển số

- Hệ thống điều khiển số



Thực hiện bộ điều khiển số

- Triển khai hàm điều khiển dưới dạng biến đổi z

$$G_R(z) = \frac{U(z)}{E(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}{a_0 + a_1 z^{-1} + \dots + a_n z^{-n}}$$

- Biểu diễn dạng rời rạc

$$a_0 u^*(t) + \sum_{k=1}^n a_k u^*(t - kT) = \sum_{k=0}^m b_k e^*(t - kT)$$

- Hoặc

$$u^*(t) = \frac{1}{a_0} \sum_{k=0}^m b_k e^*(t - kT) - \frac{1}{a_0} \sum_{k=1}^n a_k u^*(t - kT)$$

Bộ điều khiển PID số

- Phương trình bộ điều khiển PID

$$\begin{aligned}u(t) &= u_p(t) + u_i(t) + u_d(t) \\&= K \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right]\end{aligned}$$

- Biểu diễn xấp xỉ dạng rời rạc

$$u(k) = K_p \left[e(k) + \frac{T_s}{T_i} \sum_{i=0}^{k-1} e(i) + \frac{T_D}{T_s} (e(k) - e(k-1)) \right]$$

- Suy ra

$$u(k-1) = K \left[e(k-1) + \frac{T_s}{T_i} \sum_{i=0}^{k-2} e(i) + \frac{T_D}{T_s} (e(k-1) - e(k-2)) \right]$$

Bộ điều khiển PID số

- Từ hai phương trình trên ta thu được

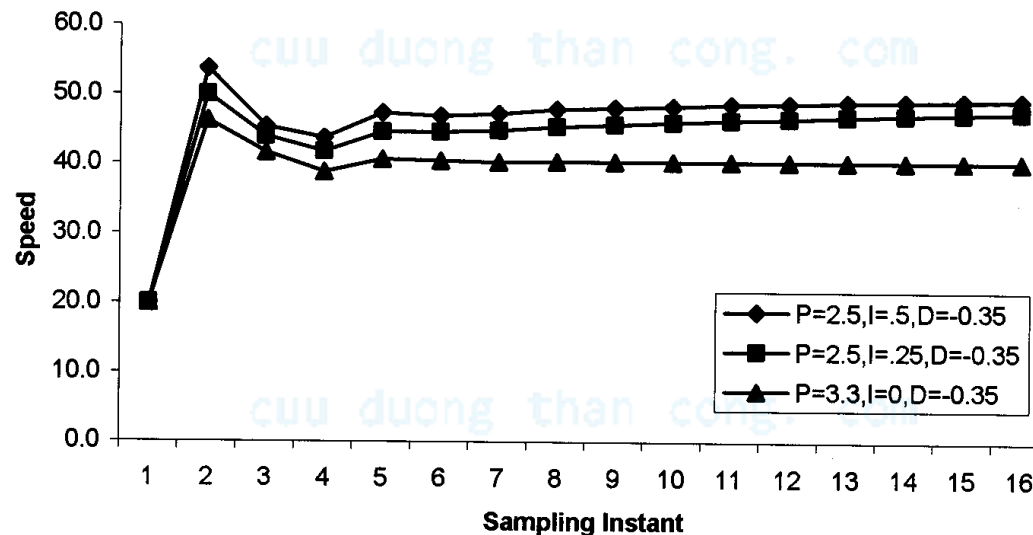
$$u(k) - u(k-1) = a_0 e(k) + a_1 e(k-1) + a_2 e(k-2)$$

trong đó, $a_0 = K \left(1 + \frac{T_D}{T_s} \right), \quad a_1 = -K \left(1 + 2 \frac{T_D}{T_s} - \frac{T_s}{T_I} \right), \quad a_2 = K \frac{T_D}{T_s}$

cuu duong than cong. com

Bộ điều khiển PID số dạng xấp xỉ

- Kết hợp điều khiển tỷ lệ, vi phân và tích phân
 - $u_t = P \cdot e_t + I \cdot (e_0 + e_1 + \dots + e_t) + D \cdot (e_t - e_{t-1})$
- Có sẵn trong công nghiệp



Mã chương trình

- Vòng lặp chương trình chính
 - Đọc đầu ra cảm biến của đối tượng điều khiển
 - Có thể yêu cầu A2D
 - Đọc đầu vào đặt hiện tại
 - Gọi hàm PidUpdate, xác định giá trị của cơ cấu chấp hành
 - Thiết lập giá trị của cơ cấu chấp hành
 - Có thể yêu cầu D2A

```
void main()
{
    double sensor_value, actuator_value, error_current;
    PID_DATA pid_data;
    PidInitialize(&pid_data);
    while (1) {
        sensor_value = SensorGetValue();
        reference_value = ReferenceGetValue();
        actuator_value =
            PidUpdate(&pid_data, sensor_value, reference_value);
        ActuatorSetValue(actuator_value);
    }
}
```

Mã chương trình (cont)

- Pgain, Dgain, Igain là hằng số
- sensor_value_previous
 - Cho khâu D
- error_sum
 - Cho khâu I

```
typedef struct PID_DATA {  
    double Pgain, Dgain, Igain;  
    double sensor_value_previous; // find the derivative  
    double error_sum; // cumulative error  
}
```

Tính toán

- $$u_t = P * e_t + I * (e_0 + e_1 + \dots + e_t) + D * (e_t - e_{t-1})$$

```
double PidUpdate(PID_DATA *pid_data, double sensor_value,
                 double reference_value)
{
    double Pterm, Iterm, Dterm;
    double error, difference;
    error = reference_value - sensor_value;
    Pterm = pid_data->Pgain * error; /* proportional term*/
    pid_data->error_sum += error; /* current + cumulative*/
    // the integral term
    Iterm = pid_data->Igain * pid_data->error_sum;
    difference = pid_data->sensor_value_previous -
                 sensor_value;
    // update for next iteration
    pid_data->sensor_value_previous = sensor_value;
    // the derivative term
    Dterm = pid_data->Dgain * difference;
    return (Pterm + Iterm + Dterm);
}
```

Dò giá trị PID

- Xác định giá trị P, I, D theo tính toán có thể không áp dụng được
 - VD không có mô hình của đối tượng, hoặc phức tạp
- Phương pháp đặc biệt để xác định giá trị P, I, D “hợp lý”
 - Bắt đầu với P, I=D=0
 - Tăng D, đến khi thấy dao động
 - Giảm D một ít
 - Tăng P, đến khi thấy dao động
 - Giảm D một ít
 - Tăng I, đến khi thấy dao động
- Lặp lại cho đến khi hết dao động

Trễ tính toán

- Luôn có trễ trong quá trình xử lý
 - Cơ cấu chấp hành tác động chậm hơn so với mong muốn
- Cần đánh giá trễ để đảm bảo có thể bỏ qua chúng mà không làm ảnh hưởng hệ thống
- Trễ phần cứng thường dễ xác định
 - Thiết kế đồng bộ
- Trễ phần mềm dễ xác định hơn
 - Tổ chức chương trình cẩn thận sao cho trễ có thể dự đoán được và tối thiểu
 - Viết chương trình có khả năng xác định thời gian
 - Watchdog timer
 - Synchronous Software Language

Lợi ích của điều khiển bằng máy tính

- Giá!!!
 - Giá thành bộ điều khiển sử dụng thiết bị tương tự đắt
 - Bộ điều khiển máy tính thay thế phần cứng analog phức tạp với mã phức tạp
- Khả năng lập trình!!!
 - Điều khiển bằng máy tính có thể được “nâng cấp”
 - Thay đổi kiểu điều khiển, chế độ điều khiển
 - Điều khiển bằng máy tính có thể thích nghi với thay đổi của đối tượng điều khiển
 - Do sự thay đổi của nhiệt độ, áp suất,...
 - “có tính mở”
 - Dễ thích nghi với các chuẩn mới