

BÁO CÁO GIỮA KÌ CƠ CẤU TRUYỀN ĐỘNG

MÔ PHỎNG ĐIỀU KHIỂN VỊ TRÍ CỦA ĐỘNG CƠ ENCODER SỬ DỤNG PID

Nguyễn Huy Thắng (22027545), Nguyễn Bảo Long (22027537), Trần Hoàng Thắng (22023160)

Tóm tắt: Mục tiêu chính của bài báo cáo là điều khiển vị trí của động cơ DC có gắn Encoder bằng cách sử dụng PID. PID sẽ hiệu chỉnh sai số giữa vị trí đã đo và vị trí mong muốn, tính toán lỗi và sau đó cấp tín hiệu điều khiển điện áp để điều chỉnh độ rộng xung để trả về kết quả phù hợp. Các tham số K_p , K_d , K_i được tinh chỉnh bằng phương pháp thử.

Từ khoá: Encoder, Quadrature Encoder, DC motor, điều khiển PID, tinh chỉnh PID

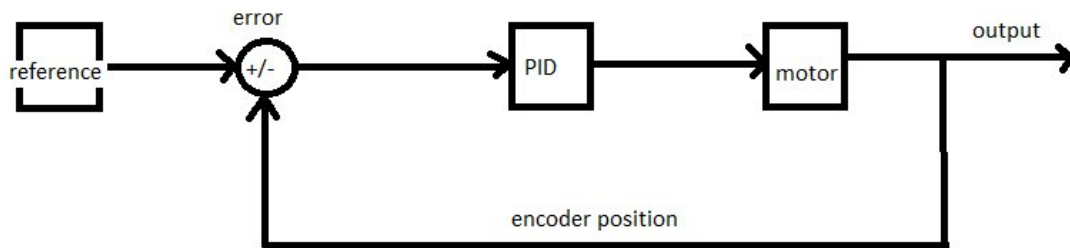
I. Giới thiệu

1. Động cơ Encoder

- Động cơ encoder là tên một loại động cơ có phần con quay được tích hợp trong một hệ thống tự động nhằm mục đích tạo tín hiệu xung trong hoạt động điều khiển động cơ hoặc máy móc.
- Encoder (bộ mã hoá) còn gọi là Quadrature Encoder, Linear Encoder, Rotary Encoder... mục đích là thực hiện mã hoá số vòng quay nhằm mục đích tạo tín hiệu xung có dạng vuông trong quá trình điều khiển động cơ.
- Dựa trên sơ đồ tín hiệu có dạng nhị phân các dòng encoder có thể hỗ trợ hoán đổi vị trí góc hoặc vị trí thẳng căn cứ theo yêu cầu sử dụng có mong muốn di chuyển đến vị trí điều khiển như thế nào.

2. Nguyên lý hoạt động

- Động cơ nhận tín hiệu điều khiển điện áp V_{app} , sinh ra dòng điện I và sinh ra mô men quay T tạo ra chuyển động quay (θ , ω).
- Bộ mã hoá encoder được gắn trực tiếp lên trục động cơ đo lường và cung cấp phản hồi thời gian thực về vị trí góc (θ , ω).
- Trong miền Laplace, hàm truyền đặc trưng của hệ thống điều khiển vị trí là từ điện áp đặt $V_{app}(s)$ đến đầu ra vị trí $\theta(s)$.
- Tín hiệu phản hồi từ encoder ($\theta(t)$) sẽ so sánh với giá trị đặt (reference) để tạo sai số $e(t)$, từ đó bộ điều khiển PID sẽ chỉnh lại điện áp điều khiển cho phù hợp.



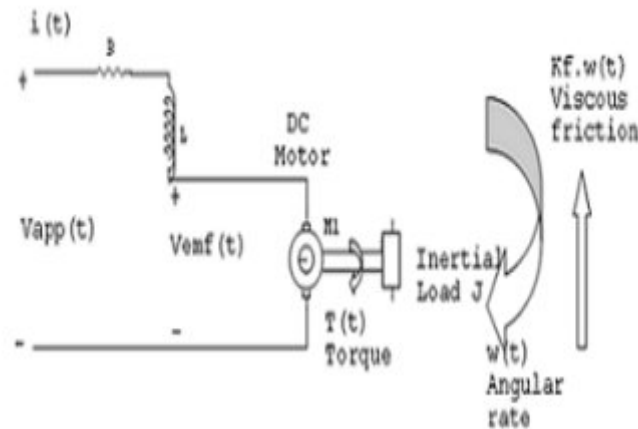
Mô hình điều khiển động cơ encoder sử dụng PID

II. Yêu cầu thiết kế

- Vị trí motor cần chính xác, sai số trạng thái cần tiến đến 0. Hệ thống cần phản hồi nhanh. Mô phỏng sử dụng reference là động cơ bước và ta hi vọng kết quả trả về có settling time khoảng 40ms và overshoot nhỏ hơn 16%.

III. Mô hình toán học và mô phỏng điều khiển

1. Mô hình toán học vị trí của động cơ DC



Cấu trúc của động cơ DC

Áp dụng định luật Kirchoff ta có:

$$V_{app} = V_L + V_R + V_{emf}$$

Trong đó:

V_{app} : điện áp cấp vào động cơ (V)

V_L : điện áp cuộn cảm (V)

V_R : điện áp trở (V)

V_{emf} : suất điện động phản kháng (V)

$$\Leftrightarrow V_{app}(t) = L \cdot \frac{di}{dt} + R \cdot i(t) + e$$

$$\Leftrightarrow V_{app}(t) = L \cdot \frac{di}{dt} + R \cdot i(t) + K_e \cdot \dot{\theta}(1)$$

Áp dụng định luật Newton ta có:

$$J \cdot \ddot{\theta} + b \cdot \dot{\theta} = T = K_t \cdot i(2)$$

Trong đó:

T : tổng mô men tác dụng lên trục quay (Nm)

J : mô men quán tính của vật quay (kgm^2)

$\ddot{\theta}$: gia tốc quay (rad/s^2)

$T_{động cơ} = K_t \cdot i$: mô men sinh ra từ động cơ điện

Áp dụng biến đổi Laplace với hai phương trình (1) và (2) ta thu được hệ phương trình:

$$(Ls + R) \cdot I(s) = V(s) - K \cdot s \cdot \theta(s)$$

$$s(Js + b) \cdot \theta(s) = K \cdot I(s)$$

Giải hệ phương trình trên ta được mối quan hệ giữa đầu vào $V(s)$ và tốc độ quay đầu ra $\theta(s)$

$$\frac{\theta(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R) + K^2}$$

2. Mô phỏng điều khiển bằng Simulink

Động cơ DC nhận điện áp $V(s)$ tạo ra mô men xoắn sinh ra chuyển động quay $\theta(s)$ từ 2 phần điện và phần cơ thông qua hàm truyền. Khối Encoder lấy ra giá trị góc quay để trả về nên khối động cơ DC motor và Encoder thay thế bằng khối hàm truyền

$$\frac{\theta(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R) + K^2}$$

Với các giá trị cho trước như sau

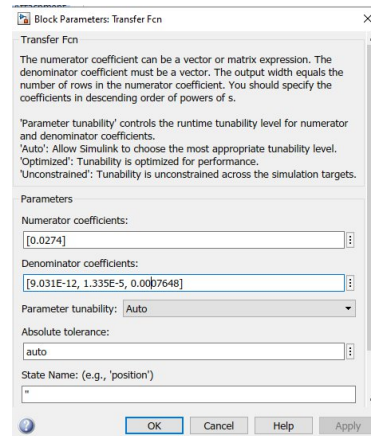
$$J = 3.284E-6;$$

$$b = 3.5077E-6;$$

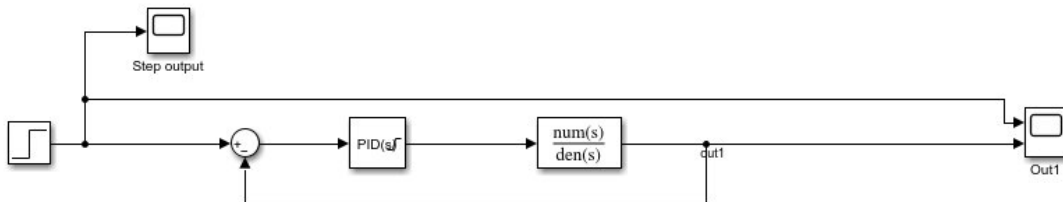
$$K = 0.0274;$$

$$R = 4;$$

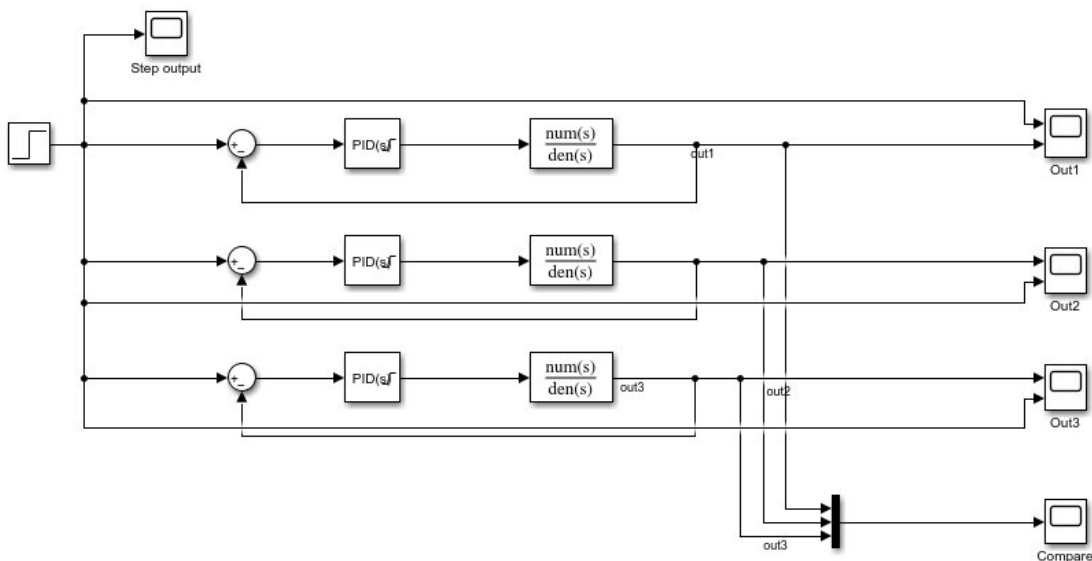
$$L = 2.75E-6;$$



Ta thiết kế bộ điều khiển động cơ có Encoder với PID trên Matlab Simulink như sau

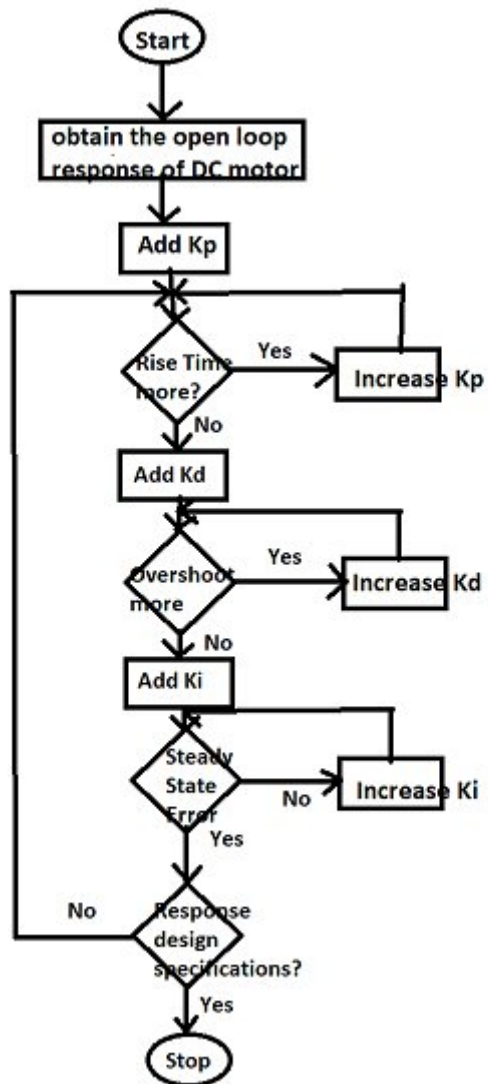


Mô hình điều khiển động cơ Encoder với PID



Mô hình với các PID khác nhau để thực hiện so sánh

Quy trình điều khiển hệ số PID có thể hiểu như sau



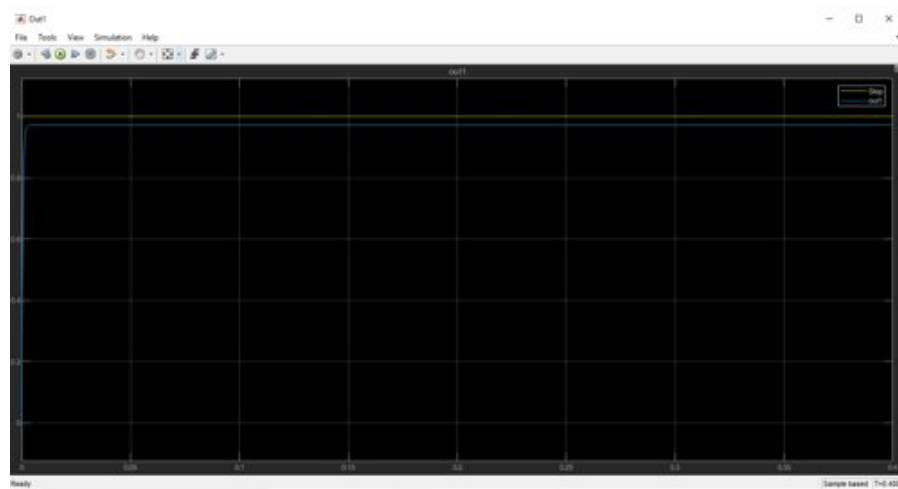
Sơ đồ tinh chỉnh hệ số PID

IV. Kết quả và nhận xét

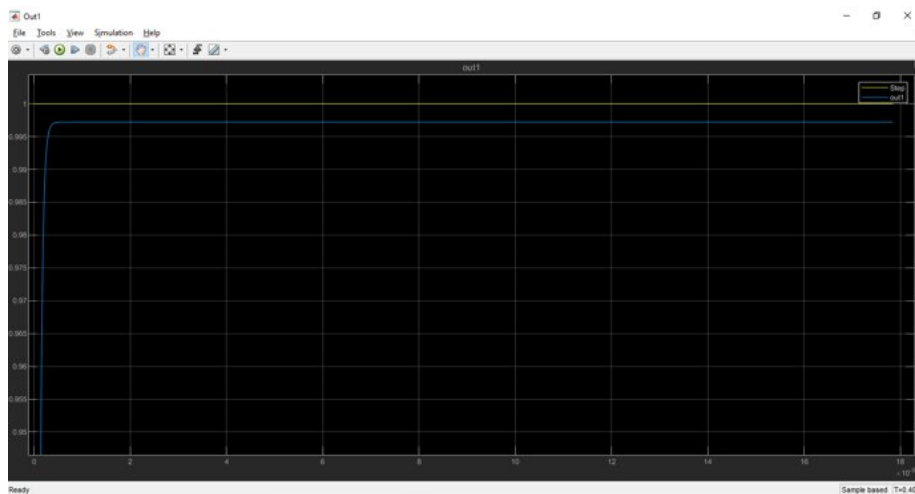
1. Kết quả



Đầu vào step

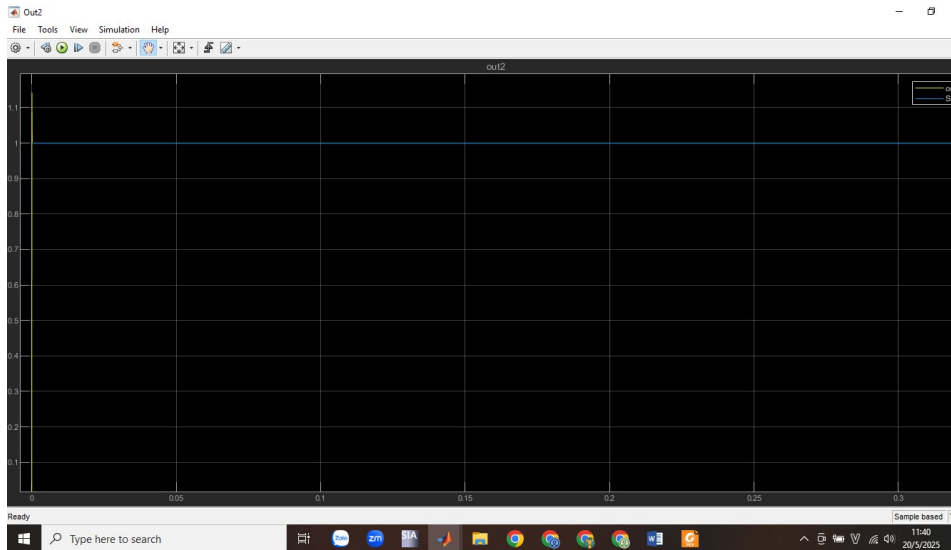


Output khi chưa có pid



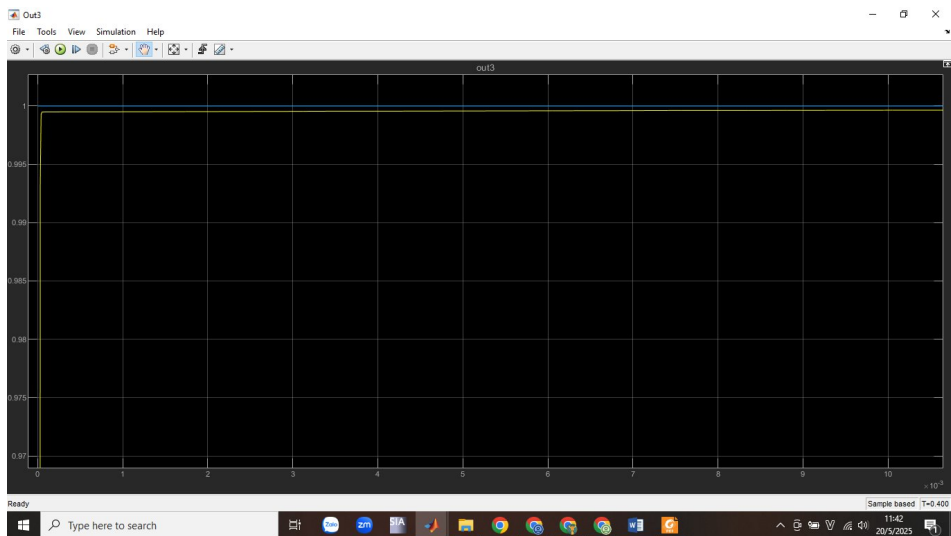
Output 1 khi đã có hệ số $K_p=10$, $K_d=K_i=0$

Khi đã có PID ta thấy cải thiện nhiều ở $\times 10^{-3}$ tuy nhiên hệ số K_p còn nhỏ nên settling time vẫn chưa như mong muốn ở Scope 1.



Output 2 khi K_p rất lớn, $K_d=K_i=0$

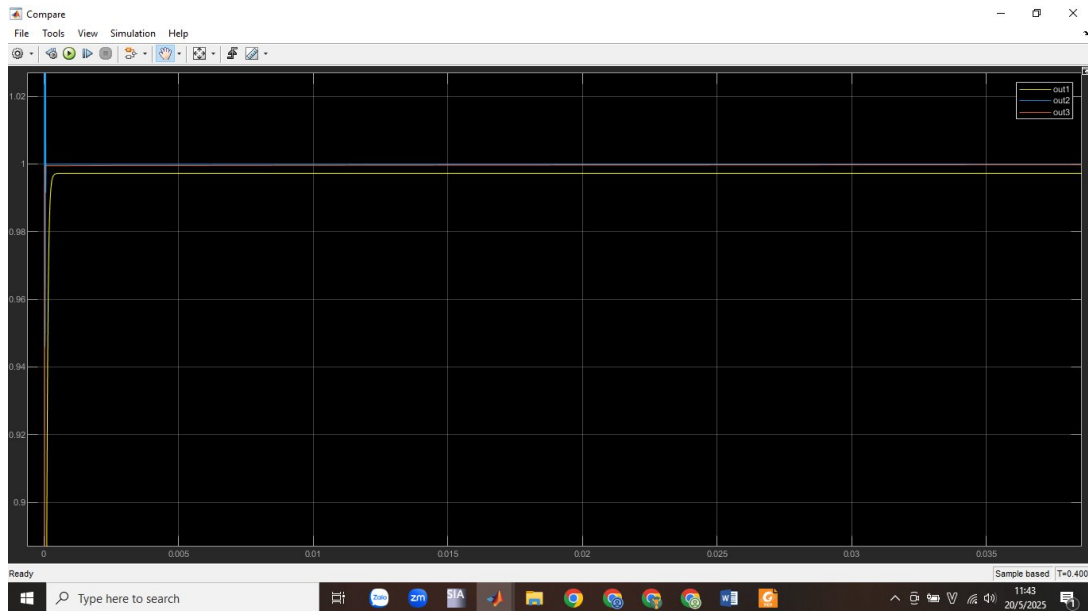
Ở scope 2 ta thấy settling time đã giảm tuy nhiên K_p quá lớn gây ra overshoot lớn ta cần tinh chỉnh thêm K_p và K_i .



Output 3 sau khi tinh chỉnh $K_p=100$, $K_d=300$, $K_i=0.35$

Sau khi tinh chỉnh cả 3 hệ số K_p , K_d , K_i ta thấy được settling time nhỏ như mong muốn, đồng thời hệ thống không bị overshoot.

Để trực quan hơn ta có thể so sánh 3 output khi tinh chỉnh các hệ số PID để so sánh bộ số PID nào là hiệu quả nhất



So sánh 3 output với 3 bộ PID

- Ⓢ Ta dễ thấy rằng line màu cam (output3) đạt độ hiệu quả nhất khi vừa chỉnh Kp đủ lớn để settling time đạt yêu cầu vừa chỉnh Kd để tránh overshoot và Ki để tránh steady state error. Tuy nhiên bộ PID trên chưa phải là tối ưu nhất.

2. Nhận xét

```
untiled.m  x  +
1 - y = out.yout.signals.values;
2 - t = out.yout.time;
3 - info = stepinfo(y, t)
4

Command Window
>> untiled

info =

struct with fields:

    RiseTime: 2.2733e-05
    SettlingTime: 2.9338e-05
    SettlingMin: 0.9932
    SettlingMax: 0.9999
    Overshoot: 0
    Undershoot: 0
    Peak: 0.9999
    PeakTime: 0.4000

fx >>
```


Hệ thống thiết kế đạt yêu cầu khi $K_p=100$, $K_i=300$, $K_d=0.35$ với RiseTime và SettlingTime rất nhỏ xấp xỉ $2.8e-05$ và Overshoot bằng 0.

Các thông số PID được chỉnh bằng phương pháp thử và kết quả mô phỏng cho thấy phương pháp trình bày đem lại kết quả tốt.

<https://github.com/thangbnsh/Position-Control-of-DC-Motor-with-Encoder-using-PID-Simulation.git>