

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN TỰ ĐỘNG



ĐỒ ÁN MÔN HỌC
THIẾT KẾ BỘ ĐIỀU KHIỂN PID SỐ
ĐỘNG CƠ DC SERVO

GVHD: BÙI THANH HUYỀN
SVTH: PHẠM TẤN THỊNH - 2014607
NGÔ QUANG AN - 2012552

TP. HỒ CHÍ MINH, THÁNG 06 NĂM 2023

LỜI CẢM ƠN

Đầu tiên, em xin được gửi lời cảm ơn chân thành nhất đến giáo viên hướng dẫn Đồ án – Cô Bùi Thanh Huyền. Cô đã là người định hướng, gợi ý về một đề tài mới, có tính chất nghiên cứu nhằm phát hiện ra một hướng tiếp cận mới trong kỹ thuật. Trong thời gian thực hiện Đồ án môn học, cô đã truyền đạt kiến thức cung như kinh nghiệm và những lời khuyên quý báu giúp em có thể hiểu rõ hơn và giải quyết vấn đề tốt hơn. Nhờ sự chỉ bảo của cô, em đã hoàn thành Đồ án cũng như nắm bắt cách thực hiện một nghiên cứu khoa học. Và đây sẽ là hành trang quý giá cho con đường phát triển sau này của em.

Em cũng xin gửi lời cảm ơn đến quý thầy cô Giảng viên Trường Đại học Bách Khoa – Đại học Quốc gia Thành phố Hồ Chí Minh, cũng như thầy cô là Giảng viên khoa Điện – Điện tử đã tận tình truyền đạt cho chúng em nhiều kiến thức quý báu.

Em đã cố gắng trình bày rõ ràng các kiến thức tìm hiểu, nghiên cứu trong đồ án. Song không thể tránh khỏi những sai sót và hạn chế. Em rất mong nhận được những ý kiến đóng góp, chỉ bảo từ quý thầy cô và các bạn.

Tp. Hồ Chí Minh, ngày 05 tháng 06 năm 2023

Sinh viên

TÓM TẮT ĐỒ ÁN

Đồ án này trình bày về sử dụng giải thuật PID ứng dụng trên vi điều khiển để điều chỉnh, ổn định động cơ DC đạt được giá trị đặt (Setpoint) mong muốn một cách nhanh nhất, tối ưu nhất cả về vị trí trực quay và tốc độ động cơ.

- Quá trình thực hiện đề tài như sau:

Xây dựng bộ điều khiển PID số, vi điều khiển STM32F103C8T6 sẽ là nhân điều khiển trung tâm. Phần công suất sử dụng module Driver L298 để điều chỉnh động cơ đạt giá trị đặt.

Sử dụng phần mềm hỗ trợ thiết kế giao diện (GUI) trên máy tính là Visual Studio, được viết bằng ngôn ngữ C# để tạo ra sự tương tác đơn giản giữ người dùng với vi điều khiển.

Cụ thể hơn, Encoder quang sẽ đưa xung phản ánh tốc độ động cơ về vi điều khiển, sau đó vi điều khiển sẽ tính toán tốc độ hiện tại của động cơ và thực hiện giải thuật điều khiển PID để điều chế độ rộng xung (PWM – Pulse Width Modulation) điều khiển động cơ thông qua Module driver L298.

Các thông số tính toán vị trí hoặc tốc độ của động cơ sẽ được truyền qua máy tính bằng chuẩn giao tiếp UART. Trên máy tính, GUI sẽ nhận các thông số trên và biểu diễn thành dạng đồ thị giúp có dễ dàng nhận biết được đáp ứng hiện tại của động cơ. Ngoài ra giao diện giúp người dùng có thể điều khiển được hệ thống, chọn loại điều khiển cũng như gửi các thông số PID và Setpoint cho vi điều khiển.

MỤC LỤC

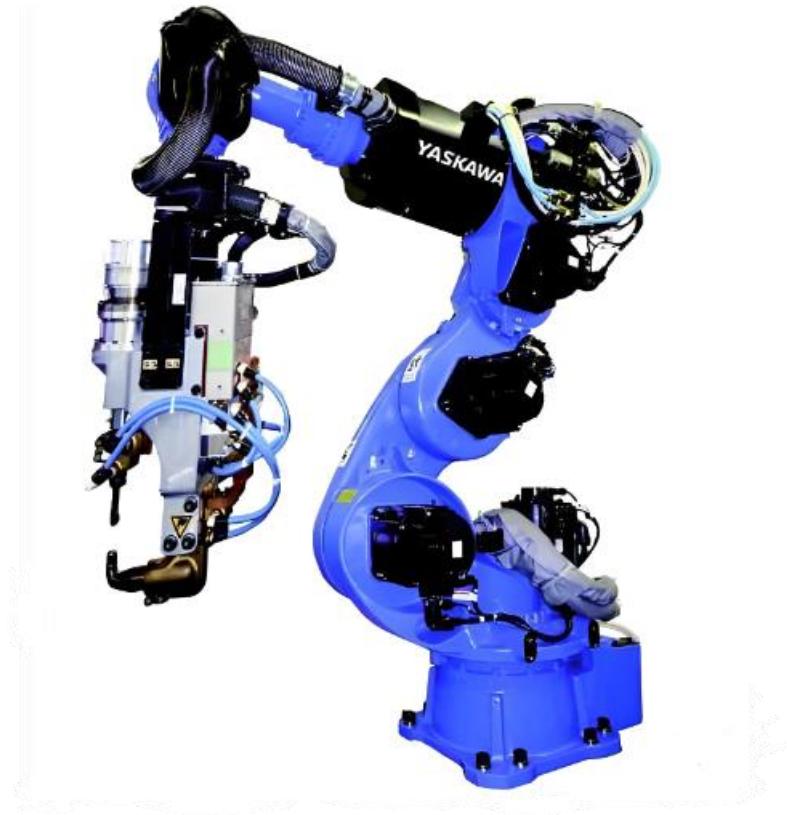
CHƯƠNG I: GIỚI THIỆU TỔNG QUAN ĐỀ TÀI	5
1.1 Đặt vấn đề	5
1.2 Hướng giải quyết	6
CHƯƠNG II: ĐỘNG CƠ ĐIỆN MỘT CHIỀU VÀ PHƯƠNG PHÁP ĐIỀU KHIỂN ĐỘNG CƠ	8
2.1 Động cơ DC	8
2.1.1 Cấu tạo và nguyên lý hoạt động	8
2.1.2 Mô hình hóa động cơ DC	9
2.2 Phương pháp điều khiển động cơ	10
CHƯƠNG III: BỘ ĐIỀU KHIỂN PID	12
3.1 Lý thuyết bộ điều khiển PID	12
3.1.1. Khâu tỉ lệ (P)	13
3.1.2. Khâu tích phân (I)	14
3.1.3. Khâu vi phân (D).....	14
3.1.4. Hiệu chỉnh bộ thông số PID	15
3.2. Bộ điều khiển PID số.....	16
CHƯƠNG IV: THIẾT KẾ VÀ THI CÔNG PHẦN CỨNG	18
4.1. Động cơ DC Servo	18
4.2. Mạch điều khiển động cơ.....	19
4.3. Vi điều khiển STM32F103C8T6	20
4.4. Nguồn cấp	22
CHƯƠNG V: LẬP TRÌNH VI ĐIỀU KHIỂN VÀ THIẾT KẾ GIAO DIỆN TRÊN MÁY TÍNH	23
5.1. Cấu trúc chương trình nhúng	23
5.1.1. Khối Read Encoder.....	23
5.1.2. Khối PID	23
5.1.3. Khối PWM.....	23
5.1.4. Khối UART Comunication	24
5.2. Sơ đồ giải thuật điều khiển của hệ thống	24
5.3. Thiết kế giao diện	24

5.3.1. Giới thiệu công cụ Visual Studio	24
5.3.2. Cấu trúc chương trình giao diện (GUI)	25
5.4. Sơ đồ các bước vận hành hệ thống trên giao diện.....	28
CHƯƠNG VI: ĐÁNH GIÁ CHẤT LƯỢNG HỆ THỐNG	29
6.1. Điều khiển vận tốc	30
6.1.1. Thay đổi K_P (Bộ điều khiển P)	30
6.1.2. Thay đổi K_I (Bộ điều khiển PI).....	32
6.1.3. Thay đổi K_D (Bộ điều khiển PD và PID).....	34
6.1.4. Nhận xét	36
6.2. Điều khiển vị trí.....	37
6.2.1. Thay đổi K_p (Bộ điều khiển P)	37
6.2.2. Thay đổi K_i (Bộ điều khiển PI).....	38
6.2.3. Thay đổi K_d (Bộ điều khiển PID).....	39
6.2.4. Nhận xét	41
CHƯƠNG VII: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	42
7.1. Kết luận	42
7.2. Hướng phát triển	44

CHƯƠNG I: GIỚI THIỆU TỔNG QUAN ĐỀ TÀI

1.1 Đặt vấn đề

Lĩnh vực điều khiển tự động ngày càng phát triển, đặc biệt là điều khiển chính xác đã trở thành một phần không thể thiếu của nền công nghiệp hiện đại. Phần lớn các loại máy móc, thiết bị dân dụng hay trong công nghiệp đều sử dụng động cơ điện như trong các máy công cụ, máy CNC, các cánh tay robot,... đến những thiết bị gia dụng như máy giặt, điều hòa, máy hút bụi, ngay cả trong máy vi tính. Những thiết bị như vậy đều có yêu cầu về độ chính xác cao, tiết kiệm năng lượng, tuổi thọ cao. Một trong những yêu cầu cần được đáp ứng đó là tiêu chí về điều khiển được tốc độ động cơ điện một cách ổn định, đáp ứng nhanh, vận hành trơn tru khi xác lập và khi thay đổi trạng thái.



HÌNH 1: CÁNH TAY ROBOT-MỘT TRONG NHỮNG ỨNG DỤNG THỰC TẾ ĐIỀU KHIỂN ĐỘNG CO

Việc ứng dụng những thuật toán kinh điển vào vấn đề điều khiển tốc độ động cơ đã đạt được nhiều kết quả khả quan. Điều khiển động cơ bằng thuật toán PID là một đề tài quen thuộc trong lĩnh vực điều khiển tự động. Dù ngày nay đã có rất nhiều giải thuật điều khiển khác được phát triển mạnh mẽ, có độ chính xác cao như điều khiển thích nghi, điều khiển mờ, thuật toán di truyền,... nhưng giải thuật PID vẫn được ứng dụng khá rộng rãi và nghiên cứu phát triển vì tính đơn giản, cũng là cơ sở để phát triển nhiều giải thuật khác.

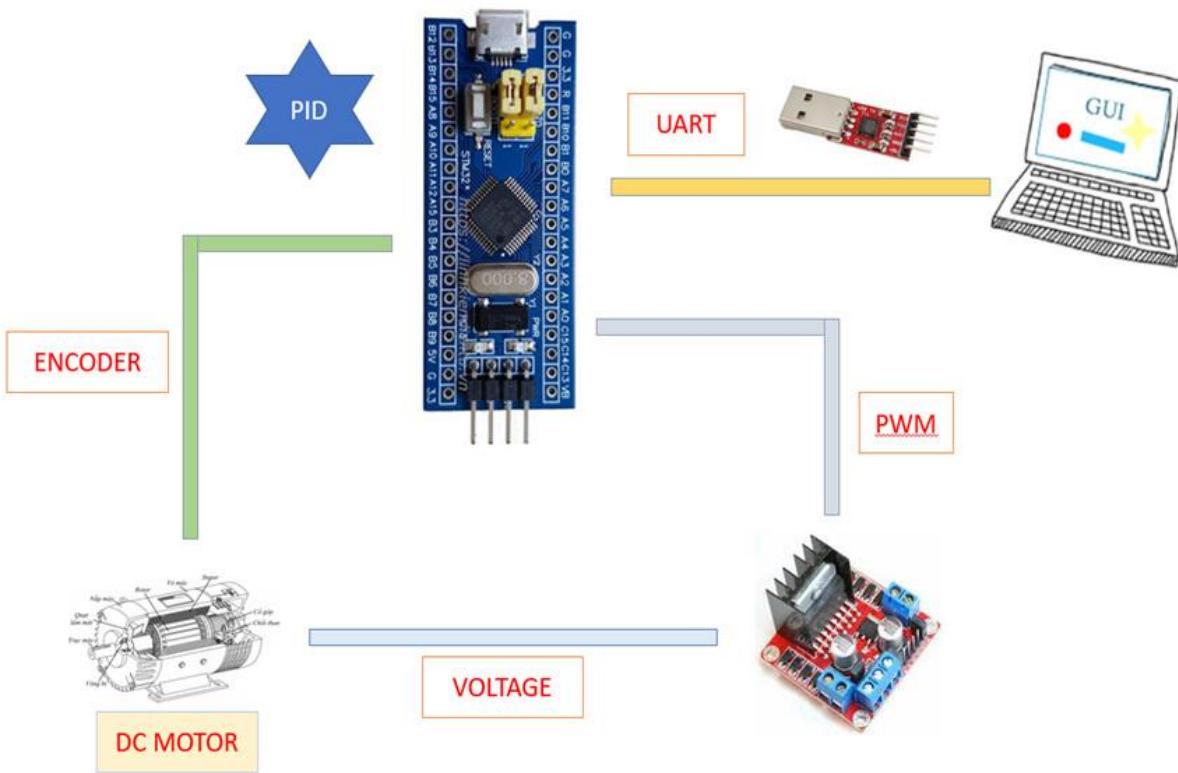
Trong khuôn khổ Đồ án, em xin trình bày về thuật toán PID và xây dựng bộ điều khiển này trên nền vi điều khiển STM32F103C8T6, sau đó ghi nhận các kết quả và hướng phát triển đề tài. Đề tài đặt ra những yêu cầu cơ bản về lý thuyết mô hình hóa đối tượng tuyến tính, giải thuật vòng kín PID, kiến thức về thiết kế phần cứng như mạch công suất, mạch nguồn. Bên cạnh đó là kỹ năng về lập trình nhúng bộ điều khiển vào vi xử lí, kỹ năng lập trình giao diện trên máy tính, tiếp cận nhiều ngôn ngữ lập trình khác nhau.

1.2 Hướng giải quyết

Tìm hiểu về động cơ một chiều (DC), các đặc tính và phương pháp điều khiển động cơ DC. Thiết kế, thi công mạch phần cứng điều khiển động cơ DC thực. Động cơ DC được sử dụng là JGB37 làm phần tử chấp hành, mạch điều khiển động cơ L298 mạch chuyển dữ liệu UART – USB CP2102 để truyền nhận dữ liệu với máy tính.

Xây dựng giải thuật và viết chương trình điều khiển, ứng dụng thuật toán điều khiển lên vi điều khiển STM32F103C8T6 làm bộ điều khiển trung tâm, sử dụng công cụ hỗ trợ lập trình STM32 CubeIDE v1.4.

Tìm hiểu cách viết giao diện trên máy tính bằng phần mềm hỗ trợ Visual Studio trên nền tảng ngôn ngữ C#.

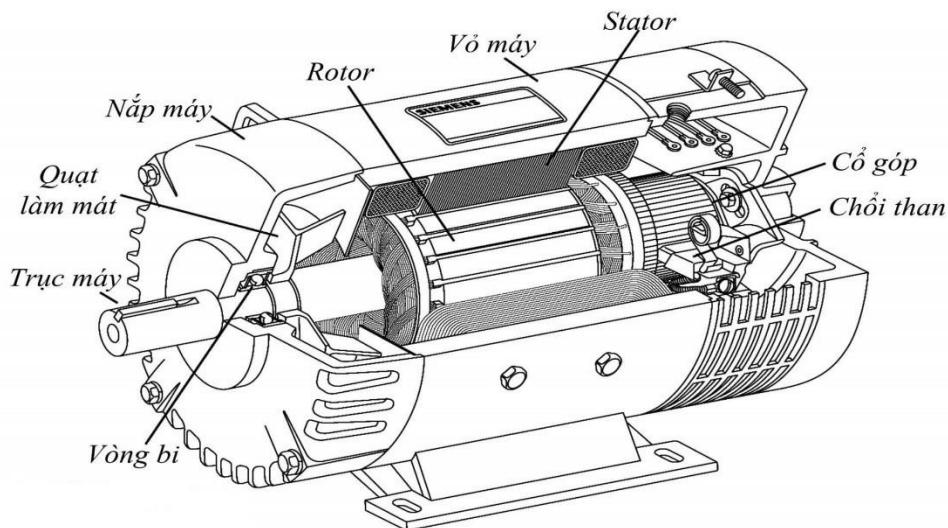


HÌNH 2: SƠ ĐỒ TỔNG THỂ HỆ THỐNG ĐIỀU KHIỂN

CHƯƠNG II: ĐỘNG CƠ ĐIỆN MỘT CHIỀU VÀ PHƯƠNG PHÁP ĐIỀU KHIỂN ĐỘNG CƠ

2.1 Động cơ DC

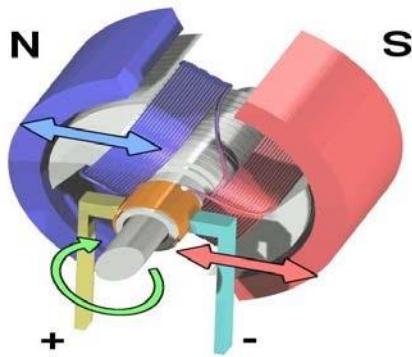
2.1.1 Cấu tạo và nguyên lý hoạt động



HÌNH 3: CẤU TẠO CỦA ĐỘNG CƠ DC

Cấu tạo của động cơ DC gồm stator, rotor, bộ chổi than – cỗ gáy:

- Stator, hay còn gọi là phần cảm của động cơ, cấu tạo gồm 2 phần chính là lõi thép và cuộn dây quấn. Ngoài ra còn có vỏ máy. Phần lõi thép Stator có dạng hình trụ với các lá thép kẽ thuật điện có dập rãnh bên trong, sau đó ghép lại tạo thành các rãnh theo hướng trục. Phần lõi này được ép vào mặt trong của vỏ máy. Cuộn dây quấn là sợi dây đồng dài được đặt trong các rãnh của lõi thép.
- Rotor, hay còn là phần ứng của động cơ. Rotor gồm lõi thép, dây quấn và trục máy. Lõi thép gồm các lá thép kẽ thuật điện được lấy từ phần bên trong của lõi thép Stator ghép lại và có mặt ngoài được dập rãnh để đặt dây quấn. Ở giữa lõi thép có dập lỗ để lắp trục động cơ.
- Bộ chổi than – cỗ gáy gồm hai lá xấp cacbon graphite thay phiên nhau quét lên hai vòng khuyên nối với nguồn điện.

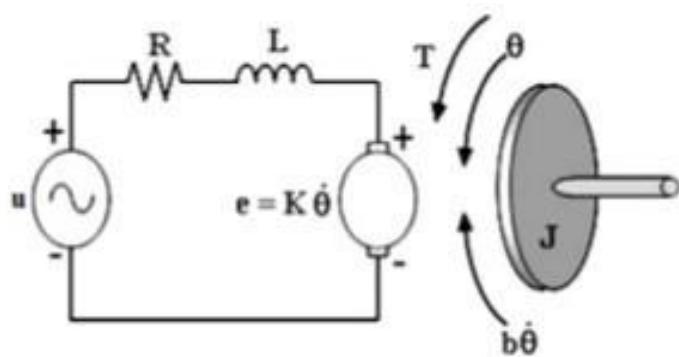


HÌNH 4: NGUYÊN LÝ HOẠT ĐỘNG ĐỘNG CƠ DC

Nguyên lý hoạt động dựa trên hiện tượng cảm ứng điện từ. Khi có dòng điện đi qua phần ứng Stator sẽ trở thành một nam châm điện tạo ra từ trường, từ đó làm xuất hiện dòng điện cảm ứng chạy trong cuộn dây rotor sinh ra lực từ Lorentz, tạo ra moment xoay trên trục động cơ. Một khi rotor quay được nửa vòng, bộ chổi than – cỗ gót đảo chiều dòng điện cấp cho rotor, giúp động cơ quay theo một hướng.

2.1.2 Mô hình hóa động cơ DC

Quan hệ điện – cơ của động cơ DC được biểu diễn như sau:



HÌNH 5: MÔ HÌNH ĐỘNG CƠ DC

R điện trở phần ứng

L điện cảm phần ứng

J momen quán tính trên trục động cơ

b hệ số ma sát

K_t hằng số momen xoắn

K_e hằng số sức phản điện

Ngõ vào điện áp nguồn U. Ngõ ra vị trí trực quay θ và tốc độ động cơ $\dot{\theta}$

Momen phát sinh trên trực động cơ: $T_m = K_t * I$ (I là dòng điện nguồn cấp)

Sức điện động trên động cơ: $E = K_e * I$

Theo định luật Newton II ta có:

$$J \frac{d^2\theta}{dt^2} + b \frac{d\theta}{dt} = T_m \quad (2.1)$$

Theo định luật Kirchhoff ta có:

$$L \frac{di}{dt} + Ri = U - K_e * \dot{\theta} \quad (2.2)$$

Biến đổi Laplace các phương trình (2.1) và (2.2) ta có:

$$\begin{aligned} s(Js + b) * \theta(s) &= K_t + I(s) \\ (Ls + R) * I(s) &= U(s) - K_e * s\theta(s) \end{aligned}$$

Từ đó, hàm truyền vị trí và tốc độ động cơ theo điện áp nguồn được tính như sau:

$$\begin{aligned} \frac{\theta(s)}{U(s)} &= \frac{1}{s} * \frac{K_t}{RJs + (bR + K_e K_t)} = \frac{1}{s} * \frac{K}{\tau s + 1} \\ \frac{\dot{\theta}(s)}{U(s)} &= \frac{K_t}{RJs + (bR + K_e K_t)} = \frac{K}{\tau s + 1} \end{aligned}$$

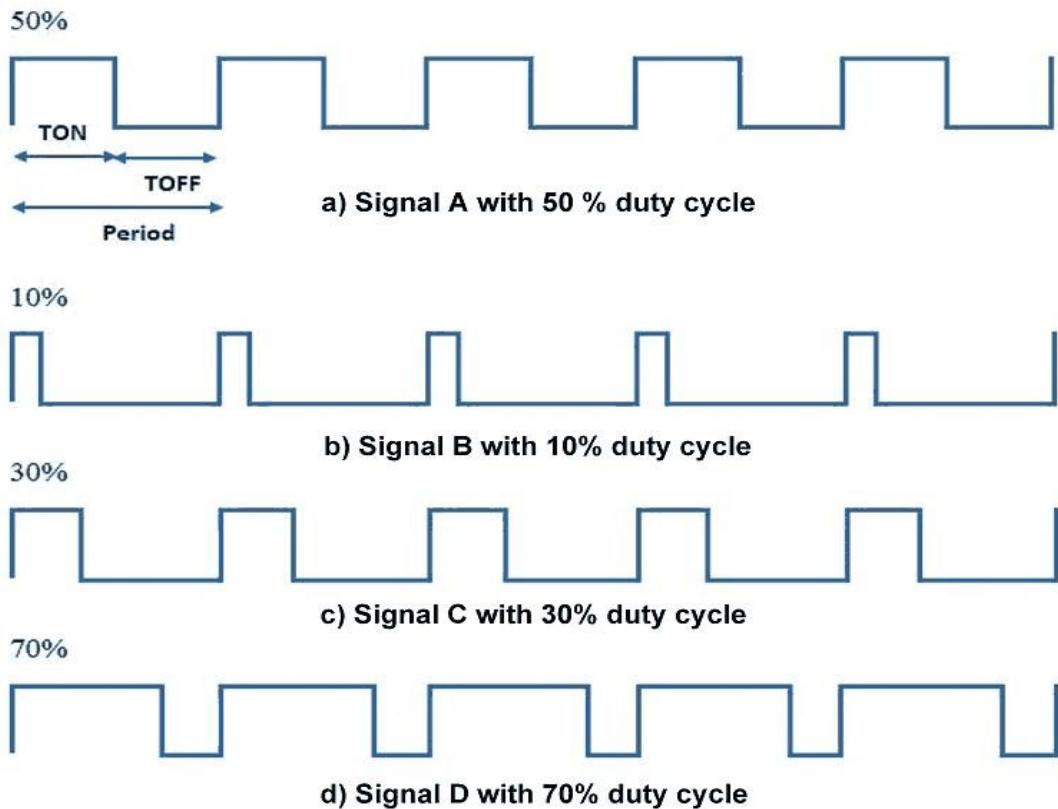
Với độ lợi và thời hằng được tính như sau:

$$K = \frac{K_t}{bR + K_e K_t} \quad \tau = \frac{JR}{bR + K_e K_t}$$

2.2 Phương pháp điều khiển động cơ

Từ hàm truyền tìm được ở trên, ta có thể thấy để thay đổi tốc độ hay vị trí trực quay, ta phải thay đổi điện áp của nguồn cấp. Tuy nhiên để thay đổi trực tiếp điện áp một chiều với tần số lớn là một điều khó khăn và gần như không thể. Do đó ta sẽ sử dụng một phương pháp tối ưu hơn đó là phương pháp điều rộng xung (PWM: Pulse Width Modulation) để thay đổi giá trị điện áp vào với tần số lớn.

PWM là phương pháp băm điện áp DC thành các xung vuông ON – OFF với tần số cao và cố định và độ rộng xung thay đổi. Do động cơ là đối tượng có quán tính, dòng điện qua động cơ không ngắt khi điện áp trong pha OFF. Do vậy việc thay đổi độ rộng xung PWM tương đương với việc thay đổi điện áp nguồn cấp.



HÌNH 6: NGUYÊN LÝ ĐIỀU CHỈ ĐỘ RỘNG XUNG PWM

Giá trị điện áp trung bình trên tải:

$$U_t = U * \frac{T_{on}}{T}$$

Với T_{on} : thời gian điện áp pha ON

T_{off} : thời gian điện áp pha OFF

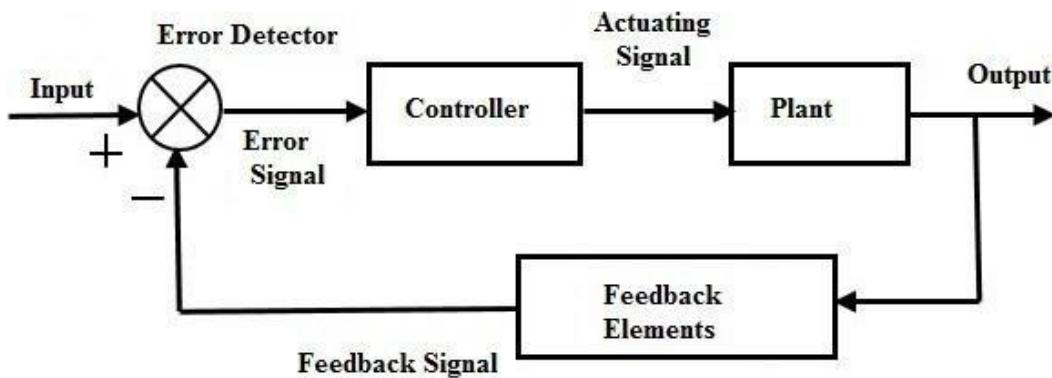
T : thời gian hết một chu kỳ xung (Period)

$$\text{Duty Cycle: } \frac{T_{on}}{T}$$

Từ công thức trên ta thấy, để thay đổi điện áp ngõ ra ta chỉ cần thay đổi Duty Cycle bằng cách thay đổi khoảng thời gian điện áp ở pha ON.

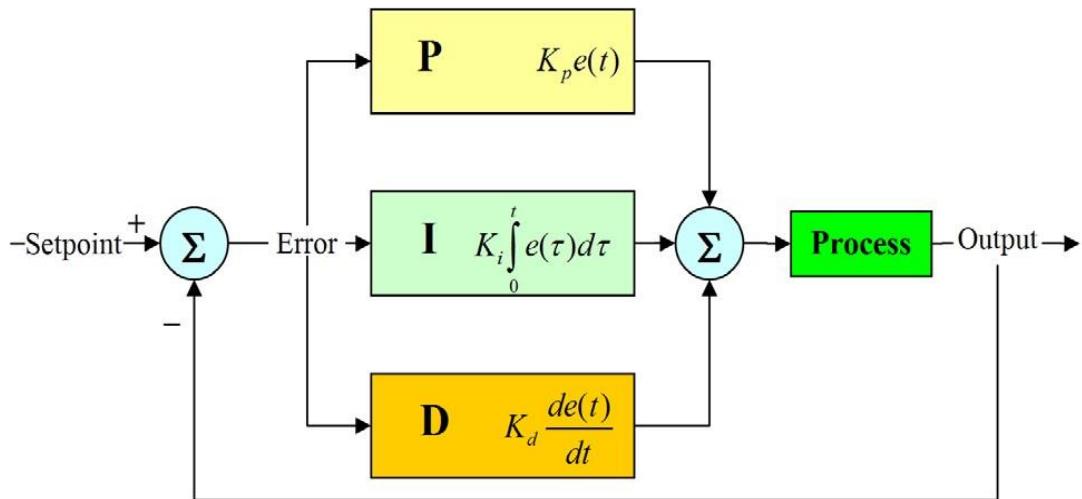
CHƯƠNG III: BỘ ĐIỀU KHIỂN PID

3.1 Lý thuyết bộ điều khiển PID



HÌNH 7: SƠ ĐỒ KHỐI CỦA HỆ THỐNG ĐIỀU KHIỂN VÒNG KÍN

Có thể nói trong lĩnh vực điều khiển, bộ điều khiển PID được xem như một giải pháp đa năng cho các ứng dụng điều khiển tương tự hay điều khiển số. Nhờ tính đơn giản và dễ dàng ứng dụng vào mọi loại đối tượng mà hơn 90% các bộ điều khiển trong công nghiệp được sử dụng là bộ điều khiển PID. Nếu được thiết kế tốt, bộ điều khiển PID có khả năng điều khiển hệ thống đáp ứng tốt các tiêu chí chất lượng khác nhau như đáp ứng nhanh, thời gian quá độ ngắn, độ vọt lô thấp, triệt tiêu được sai lệch tĩnh.



HÌNH 8: SƠ ĐỒ KHÓI BỘ ĐIỀU KHIỂN PID

Luật điều khiển PID được định nghĩa:

$$u(t) = K_p \left(e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right) \quad (3.1)$$

Trong đó, u là tín hiệu điều khiển và e là sai lệch điều khiển ($e = \text{input} - \text{output}$). Tín hiệu u là tổng của ba thành phần: khâu tỉ lệ, khâu tích phân và khâu vi phân.

Hàm truyền của bộ điều khiển PID:

$$G_{PID}(s) = K_p + K_i \frac{1}{s} + K_d s \quad (3.2)$$

Các tham số của bộ điều khiển là K_p , K_i và K_d .

3.1.1. Khâu tỉ lệ (P)

$$u(t) = K_p e(t)$$

Tác động của thành phần tích đơn giản là tín hiệu điều khiển tỉ lệ tuyến tính với sai lệch điều khiển. Ban đầu, khi sai lệch lớn thì tín hiệu điều khiển sẽ lớn. Sai lệch giảm dần thì tín hiệu điều khiển cũng giảm dần. Khi sai lệch $e(t) = 0$ thì $u(t) = 0$. Một vấn đề là khi sai lệch đổi dấu thì tín hiệu cũng sẽ đổi dấu.

Thành phần P có ưu điểm là tác động nhanh và đơn giản. Hệ số tỉ lệ K_p càng lớn thì tốc độ đáp ứng càng nhanh, do đó thành phần P có vai trò lớn trong giai đoạn đầu của quá trình quá độ.

Tuy nhiên, khi hệ số tỉ lệ K_p càng lớn thì sự thay đổi tín hiệu điều khiển sẽ càng mạnh dẫn đến dao động lớn, đồng thời làm hệ nhạy cảm hơn với nhiễu do vì nhiễu cũng sẽ bị khuếch đại theo hệ số K_p .

3.1.2. Khâu tích phân (I)

$$u(t) = K_I \int_0^t e(\tau) d\tau$$

Với thành phần tích phân, khi tồn tại một sai lệch điều khiển dương, luôn làm tăng tín hiệu điều khiển và khi sai lệch là âm thì luôn làm giảm tín hiệu điều khiển, bất kể sai lệch đó có giá trị lớn hay nhỏ. Do đó, ở trạng thái xác lập, sai lệch bị triệt tiêu $e(t) = 0$. Đây cũng là ưu điểm của thành phần tích phân.

Nhược điểm của thành phần tích phân là do phải mất một khoảng thời gian đợi $e(t)$ về 0 nên đặc tính tác động của bộ điều khiển sẽ chậm hơn xác lập hơn. Ngoài ra, thành phần tích phân đôi khi còn làm xấu đi đặc tính động học của hệ thống, thậm chí có thể làm mất ổn định nếu hệ số K_I quá lớn, hệ sẽ liên tục dao động đảo chiều liên tục trong khoảng thời gian ngắn.

Người ta thường sử dụng bộ PI hoặc PID thay vì chỉ bộ I đơn thuần vừa để cải thiện tốc độ đáp ứng, vừa đảm bảo yêu cầu động học.

3.1.3. Khâu vi phân (D)

$$u(t) = K_D \frac{de(t)}{dt}$$

Mục đích của thành phần vi phân là cải thiện sự ổn định của hệ kín. Do động học của quá trình nên sẽ tồn tại một khoảng thời gian trễ làm bộ điều khiển chậm so với sự thay đổi

của sai lệch $e(t)$ và ngõ ra output của quá trình. Khâu vi phân đóng vai trò dự đoán đầu ra của quá trình và đưa ra phản ứng thích hợp dựa trên chiều hướng và tốc độ thay đổi của sai lệch $e(t)$, làm tăng tốc độ đáp ứng của hệ. Một ưu điểm khác là thành phần vi phân giúp ổn định các quá trình mà bình thường không ổn được của các bộ P hay PI.

Nhược điểm của khâu vi phân là rất nhạy cảm với nhiễu đo hay của giá trị đặt do tính đáp ứng nhanh của nó.

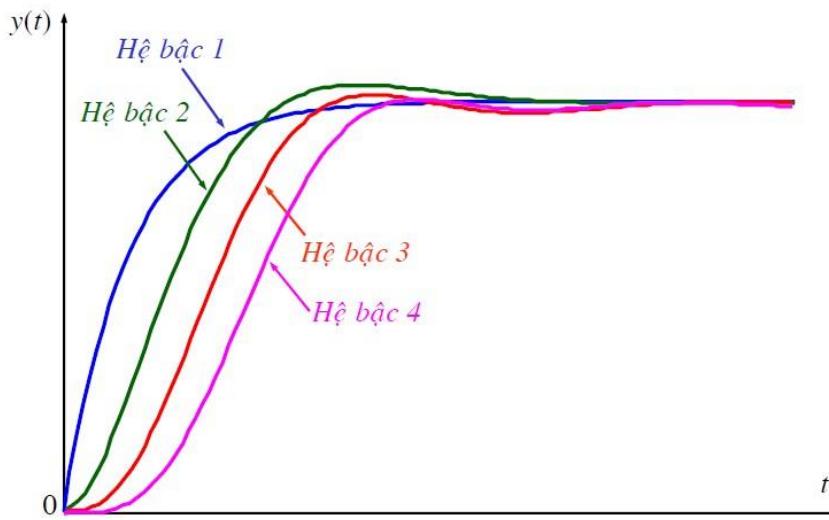
3.1.4. Hiệu chỉnh bộ thông số PID

Do từng thành phần của bộ PID có những ưu nhược điểm khác nhau và không thể đồng thời đạt được tất cả các tiêu chí chất lượng một cách tối ưu, nên việc lựa chọn thông số sẽ quyết định đến chất lượng đầu ra của hệ thống, và nó được thỏa hiệp giữa các yêu cầu chất lượng và mục đích điều khiển đặt ra ban đầu. Việc lựa chọn tham số cho bộ điều khiển PID cũng phụ thuộc vào đối tượng điều khiển và các phương pháp xác định thông số. Tuy nhiên, kinh nghiệm cũng là một yếu tố quan trọng.

	Thời gian lén	Vọt lô	Thời gian quá độ	Sai số xác lập	Độ ổn định
K_P	Giảm	Tăng	Thay đổi nhỏ	Giảm	Giảm
K_I	Giảm	Tăng	Tăng	Loại bỏ	Giảm
K_D	Thay đổi nhỏ	Giảm	Giảm	Không tác động	Cải thiện

Bảng 1: Ảnh hưởng của các hệ số đến đáp ứng ngõ ra của hệ thống

- Để có thể tối ưu hóa đáp ứng xác lập, ta chỉnh các hệ số sao cho sai số xác lập bằng 0.
- Để tối ưu hóa đáp ứng quá độ, ta chỉnh sao cho đáp ứng có vọt lô thấp, thời gian lén và thời gian quá độ ngắn. Tiêu chuẩn ITAE (Integral of Time multiplied by the Absolute Value of the Error – tích phân của thời gian nhân với trị tuyệt đối của sai số) giúp ta tham khảo về chất lượng quá độ.



HÌNH 9: ĐÁP ỨNG PID TỐI UU THEO CHUẨN ITAE

3.2. Bộ điều khiển PID số

Trong thực tế công nghiệp, các bộ điều khiển PID có thể được cấu thành từ các mạch tương tự hoặc các cơ cấu chấp hành. Tuy nhiên với yêu cầu cao về độ chính xác và chống nhiễu tốt thì các bộ điều khiển như vậy chưa đáp ứng được. Tuy nhiên với sự phát triển nhanh chóng của các ứng dụng nhúng trên nền vi xử lý, thì điều khiển số đã ra đời để giúp giải quyết vấn đề này. Các bộ điều khiển được số hóa để có thể thực thi với tốc độ cao và chính xác hơn. Đồng thời việc xây dựng các bộ điều khiển trên nền máy tính số hay vi điều khiển cũng đơn giản hơn nhiều.

Quá trình biến đổi xấp xỉ bộ PID trên miền thời gian sang dạng PID số được thực hiện như sau, lưu ý ta cần quan tâm đến tham số trọng tâm nhất là chu kỳ lấy mẫu của vi điều khiển:

Từ công thức (3.1):

$$u(t) = K_p \left(e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{de(t)}{dt} \right)$$

Ta sử dụng các công thức xấp xỉ tích phân lùi (backward integral approximation) và vi phân lùi (backward difference approximation) với chu kỳ lấy mẫu là Ts.

$$\int_0^t e(\tau) d\tau \approx \sum_{k=1}^n T_s * e(kT_s)$$

$$\frac{de(t)}{dt} \approx \frac{e(kT_s) - e(kT_s - T_s)}{T_s}$$

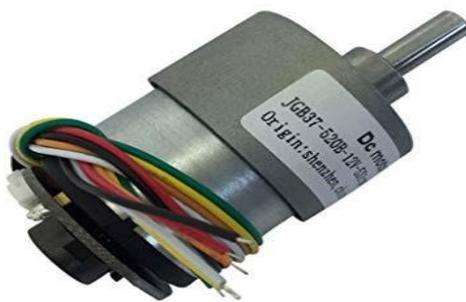
Khi đó, ta sẽ có công thức PID số như sau:

$$u(kT_s) = K_p \left(e(kT_s) + \frac{T_s}{T_i} \sum_{k=1}^n T_s * e(kT_s) + T_D \frac{e(kT_s) - e(kT_s - T_s)}{T_s} \right)$$

CHƯƠNG IV: THIẾT KẾ VÀ THI CÔNG PHẦN CỨNG

4.1. Động cơ DC Servo

Trong khuôn khổ Đồ án này, phần tử chấp hành được sử dụng là động cơ DC Servo có giảm tốc. Động cơ có tích hợp đặc điểm kĩ thuật tiện ích là Encoder, hỗ trợ việc đọc chính xác tốc độ động cơ, hộp số cho phép tăng moment xoắn qua đó tăng khả năng mang tải của động cơ.

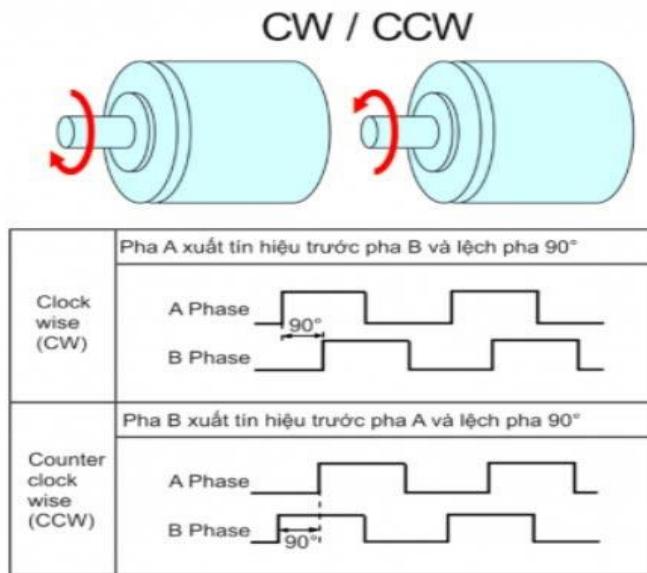


Hình 10: Động cơ DC JGB37 520 1:30

Thông số kĩ thuật	
Động cơ	
Điện áp định mức	6 -24V/ 12VDC
Dòng tối đa	3A
Công suất	36W
Tốc độ không tải (trước giảm tốc)	9990 rpm
Tốc độ không tải (sau giảm tốc)	333 rpm
Tỉ lệ Hộp số	1:30
Encoder	
Điện áp nguồn	3.3 - 5VDC
Số xung/vòng (trước giảm tốc)	11
Số xung/vòng (sau giảm tốc)	330

Bảng 2: Thông số kĩ thuật của động cơ DC JGB37 520 1:30

Dựa vào encoder ta có thể xác định được chiều quay của động cơ. Qui định xung A xuất tín hiệu trước xung B thì encoder đang quay cùng chiều kim đồng hồ CW(Clockwise), lúc này xung A đi trước và lệch pha 90 độ so với xung B. Ngược lại trực encoder quay ngược chiều kim đồng hồ CCW(Counter Clockwise), thì xung B xuất tín hiệu trước và lệch pha 90 độ xung A.



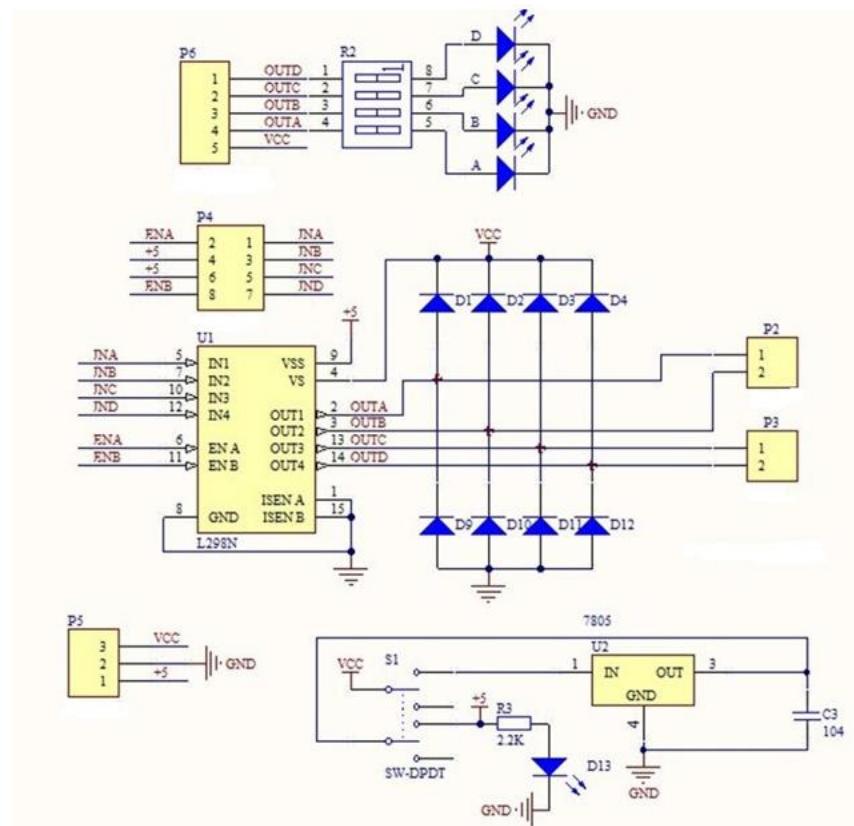
Hình 11: Cách xác định chiều quay của động cơ

4.2. Mạch điều khiển động cơ

Mạch điều khiển động cơ là thiết bị công suất dành để cấp nguồn cho động cơ theo sự biến đổi của tín hiệu điều khiển. Đồ án chọn sử dụng module driver L298 với nhiều ưu điểm như có thời gian giãn cách khi đổi chiều, công suất phù hợp với động cơ nhỏ, giá thành thấp, sử dụng đơn giản.



Hình 12: Module driver L298



Hình 13: Sơ đồ nguyên lý module driver

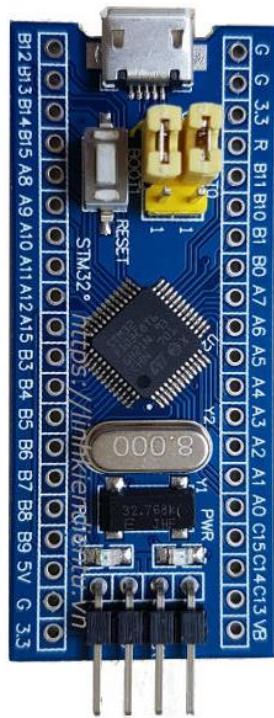
Thông số kỹ thuật	
Số cầu	2
Điện áp nguồn	5 – 30 VDC
Ngõ vào tín hiệu	5 VDC
Công suất	25W/1 cầu
Dòng ngõ ra tối đa	2A

Bảng 3: Thông số kỹ thuật của module driver L298

4.3. Vi điều khiển STM32F103C8T6

Để thực hiện được việc thiết kế bộ điều khiển PID điều khiển động cơ, ta sẽ dùng bộ xử lý trung tâm là STM32F1 với nhân chính là chip xử lý STM32F103C8T6; là dòng vi xử lý 32-bit lõi ARM Cortex-M3. Việc sử dụng board này sẽ giúp thúc đẩy khả năng của vi điều khiển làm việc với hiệu suất tốt nhất, nó cho phép người dùng dễ dàng phát triển các

ứng dụng khác nhau. Kit bao gồm một mạch nạp ST-LINK/V2 để phục vụ việc nạp code và debug chương trình, cảm biến gia tốc ST-MEMS, một micro sô, một audio DAC với trình điều khiển loa tích hợp lớp D, đèn LED, nút bấm và cổng micro-AB USB OTG.



Hình 14: Kit STM32F1038T6

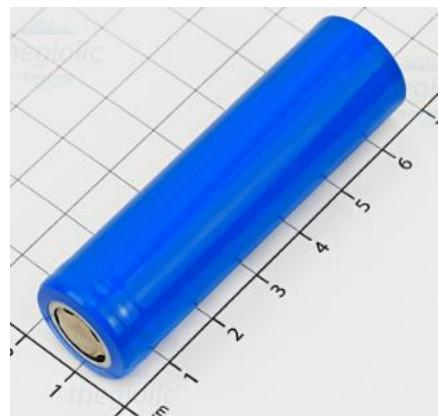
Thông số kỹ thuật	
RAM	20KB
Flash memory	64Kb
Nguồn cấp qua cổng USB	5 VDC
Nguồn cấp ngoài	3VDC, 5VDC
Thạch anh	4 – 16MHz
Đao động RC	8MHz
ADC	2 bộ 12 bit với 9 kênh mỗi bộ
DMA	7 kênh
Timer	4 kênh – 16 bit

I2C	2 kênh
USART/UART interface	3 kênh
SPI interface	2 kênh
CAN interface	1 kênh
USB interface	USB 2.0 full-speed

Bảng 4: Thông số kỹ thuật của kit STM32F103C8T6

4.4. Nguồn cấp

Để thuận tiện cho di chuyển cũng như độ linh động gọn nhẹ, trong đồ án sẽ sử dụng nguồn cấp là PIN 18650 (3 viên). Dung lượng định mức của một pin là 4200mAh, điện áp định mức 3.7V, dòng xả 2A ~ 8A.

*Hình 15: Pin 18650*

CHƯƠNG V: LẬP TRÌNH VI ĐIỀU KHIỂN VÀ THIẾT KẾ GIAO DIỆN TRÊN MÁY TÍNH

5.1. Cấu trúc chương trình nhúng

Vi điều khiển đóng vai trò là bộ điều khiển trung tâm của hệ thống. Để có thể lập trình cho vi điều khiển ta sử dụng phần mềm hỗ trợ STM32 CubeIDE v1.4 viết bằng thư viện HAL (Hardware Abstraction Layer). Chương trình của vi điều khiển gồm các khối chức năng sau: Read Encoder, PID Algorithm, PWM Generator. Để thuận tiện, dễ dàng giao tiếp với máy tính, chương trình sẽ có thêm khối UART Communication.

5.1.1. Khối Read Encoder

Khối này có chức năng đọc tín hiệu Encoder bằng mode đọc encoder được hỗ trợ sẵn trong chip sau đó sẽ chuyển thành dữ liệu vị trí trực quay với đơn vị độ (degree) và tốc độ động cơ theo đơn vị vòng/phút (rpm). Ở ngõ ra của khối sẽ có bộ lọc thông thấp để hạn chế nhiễu do sai số của Encoder và các xung đột ngọt sinh ra do động cơ bị giật ngược khi đảo chiều quay.

5.1.2. Khối PID

Khối PID nhận tín hiệu đặt (Setpoint), các thông số bộ điều khiển từ giao diện trên máy tính thông qua khối UART Communication, đồng thời nhận tín hiệu hồi tiếp vị trí, tốc độ của khối Read Encoder, sau đó sẽ áp dụng giải thuật PID tính toán để tìm được tín hiệu điều khiển.

5.1.3. Khối PWM

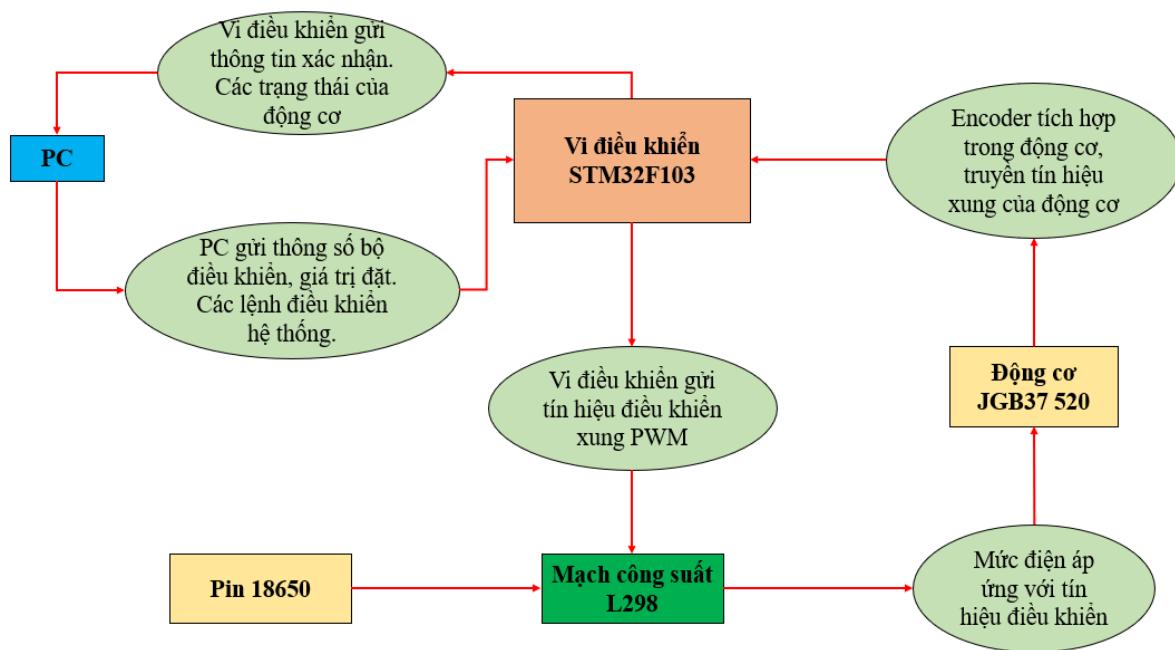
Khối PWM nhận tín hiệu điều khiển từ khối PID và tiến hành xuất xung PWM tương ứng với tín hiệu điều khiển nhận được. Ở ngõ vào của khối sẽ có bộ bão hòa để số xung PWM phát ra không vượt quá số xung tối đa trong một chu kì, cũng như để đảm bảo an toàn cho mạch Driver không vượt quá 90% công suất.

Ngõ ra sẽ có 2 tín hiệu xuất xung PWM tương ứng với trường hợp quay thuận và nghịch của động cơ. Khi tín hiệu điều khiển là dương thì tín hiệu xung ở chân 1 sẽ là số xung PWM và tín hiệu chân số 2 sẽ bằng 0, và ngược lại đối với tín hiệu điều khiển âm.

5.1.4. Khối UART Comunication

Khối này nhận tín hiệu đặt và thông số bộ điều khiển từ máy tính, rồi gửi cho khói PID để tính toán tín hiệu điều khiển. Bên cạnh đó, khói UART sẽ lấy dữ liệu từ khói đọc encoder rồi gửi lên máy tính để giao diện thực hiện vẽ đồ thị đáp ứng của động cơ theo thời gian thực, giúp ta có thể dễ dàng theo dõi, giám sát quá trình hoạt động của động cơ cũng như đánh giá chất lượng của bộ điều khiển.

5.2. Sơ đồ giải thuật điều khiển của hệ thống



5.3. Thiết kế giao diện

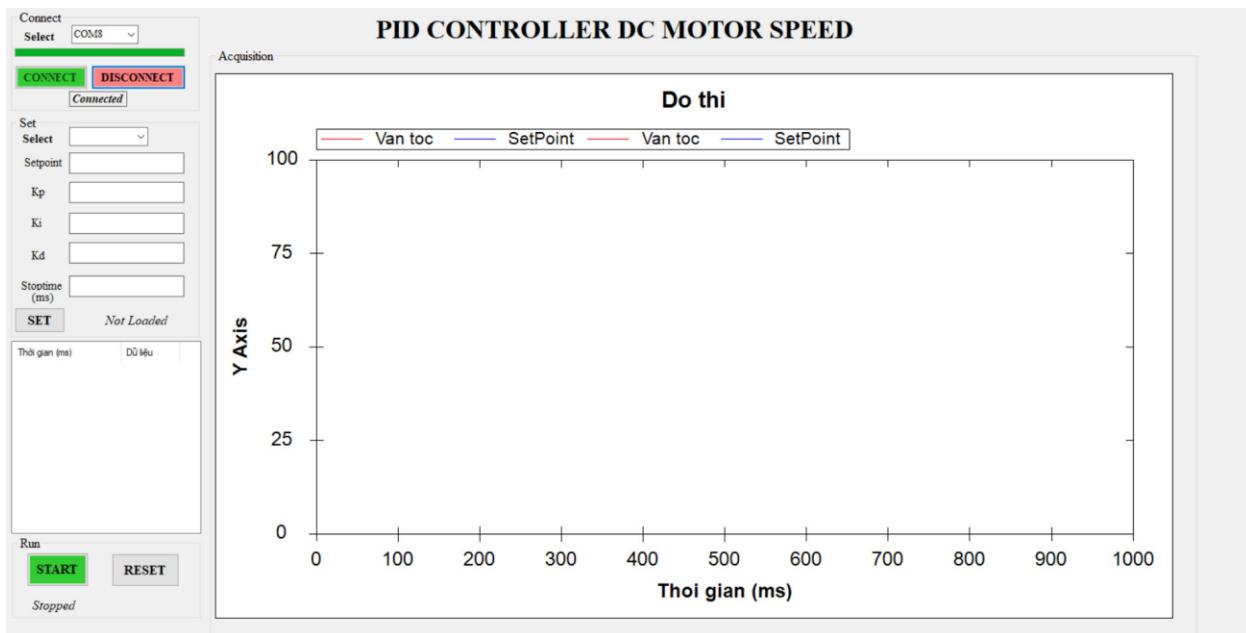
5.3.1. Giới thiệu công cụ Visual Studio

Giao diện được lập trình bằng phần mềm Visual Studio trên hệ điều hành Windows. Đây là một công cụ quen thuộc với sinh viên ngành điện và máy tính, hỗ trợ tốt cho việc

giao tiếp máy tính. Visual Studio có giao diện dễ dùng, hiệu năng cao và chiếm ít tài nguyên máy.

5.3.2. Cấu trúc chương trình giao diện (GUI)

Giao diện có các khối kết nối, khối cài thông số, khối hiển thị dữ liệu nhận về, khối lệnh và khối vẽ đồ thị.



Hình 16: Cấu trúc giao diện tương tác trên máy tính

5.3.2.1. Khối kết nối

Có chức năng chọn cổng COM kết nối với USB UART đang kết nối giữa máy tính và vi điều khiển, có nút nhấn để cho phép kết nối cũng như ngắt kết nối giữa 2 thiết bị. Khi chưa kết nối nút nhấn sẽ hiển thị chữ “CONNECTED” màu chữ xanh lá và khi đã kết nối thành công thì sẽ hiển thị chữ “DISCONNECTED” màu chữ đỏ để báo cho người sử dụng biết. Khi đang trong kết nối, bảng chọn cổng COM sẽ bị khóa lại. Khi đang không kết nối, các nút nhấn trong System Control cũng như nút gửi thông số bộ điều khiển sẽ bị khóa lại.

5.3.2.2. Khối lệnh

Khối này có chức năng gửi các lệnh điều khiển cho phép hệ thống chạy, dừng, hoặc đặt hệ thống về lại trạng thái mặc định.

Khi nhấn nút “START”, máy tính sẽ gửi xuống vi điều khiển một khung tin. Sau khi xác nhận được khung tin, vi điều khiển sẽ gửi trả một khung tin cho máy tính xác nhận động cơ đang chạy và sẽ bắt đầu khởi động bộ điều khiển và xuất xung PWM. Máy tính nhận được khung tin sẽ hiển thị thông báo “STARTED”, và bắt đầu nhận dữ liệu. Để có thể chạy được, hệ thống phải có đầy đủ thông số PID và chọn mode điều khiển.

Khi nhấn nút “STOP”, máy tính sẽ gửi xuống vi điều khiển một khung tin. Sau khi xác nhận được khung tin, vi điều khiển sẽ gửi trả một khung tin cho máy tính xác nhận đã ngừng lại động cơ, và sẽ ngắt bộ điều khiển PID và xuất xung PWM = 0 để dừng động cơ, đồng thời đưa biến đọc encoder về giá trị 0. Máy tính nhận được khung tin sẽ hiển thị thông báo “STOPPED”.

Khi nhấn nút “RESET”, máy tính sẽ gửi xuống vi điều khiển một khung tin. Sau khi xác nhận được khung tin, vi điều khiển dừng động cơ (nếu đang chạy), đặt lại các khối PID, PWM, Encoder về ban đầu và gửi trả một khung tin cho máy tính xác nhận đã đưa hệ thống về trạng thái mặc định. Máy tính nhận được khung tin sẽ hiển thị thông báo “RESETTED” và làm mới lại các khối đồ thị và khối hiển thị dữ liệu.

5.3.2.3. Khối cài đặt thông số

Có chức năng chọn chế độ điều khiển vận tốc hay vị trí, đặt tín hiệu đặt, đặt các thông số của bộ điều khiển PID và thời gian kết thúc một lần chạy.

Khi nhấn nút “SET”, máy tính sẽ gửi xuống vi điều khiển một khung tin. Khi nhận đủ cài đặt, vi điều khiển sẽ gửi trả lại một khung tin xác nhận. Lúc này, việc cài đặt hoàn tất, máy báo “LOADED”. Nếu có thay đổi mới trong cài đặt, máy báo “NOT LOADED”.

Cần thiết lập kết nối trước khi có thể gửi cài đặt xuống vi điều khiển.

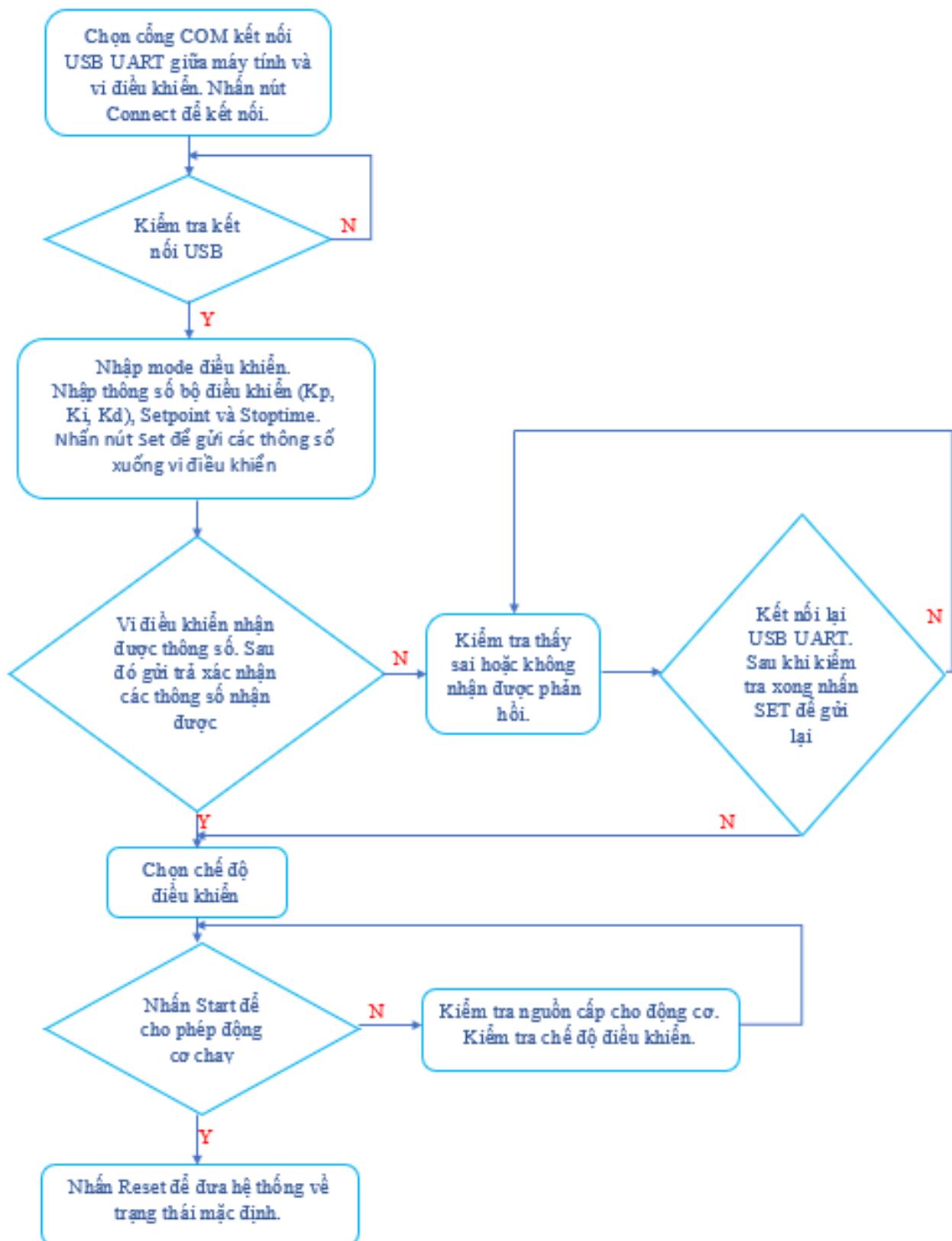
5.3.2.4. Khối vẽ đồ thị

Khối vẽ đồ thị phụ thuộc vào các khối lệnh. Khi nhấn “START”, khối xóa đồ thị cũ. Khi vi điều khiển gửi dữ liệu mỗi 10 mili giây, đồ thị được vẽ thêm. Nếu dữ liệu có giá trị vượt khỏi thang đo hiện tại của đồ thị, nó sẽ tự điều chỉnh tầm rộng hơn và vẽ lại dữ liệu trên thang mới. Khi hết thời gian chạy, đồ thị dừng vẽ. Khi nhấn “RESET”, đồ thị được xóa.

5.3.2.5. Khối nhận dữ liệu

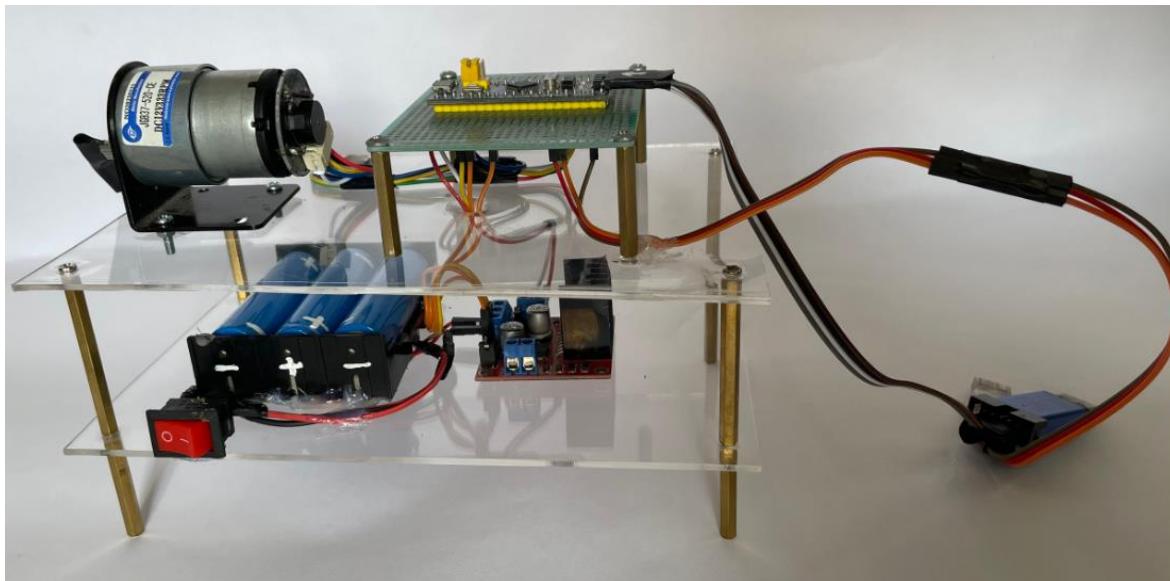
Khối này chủ yếu làm nhiệm vụ hiển thị giá trị hiện tại của động cơ.

5.4. Sơ đồ các bước vận hành hệ thống trên giao diện

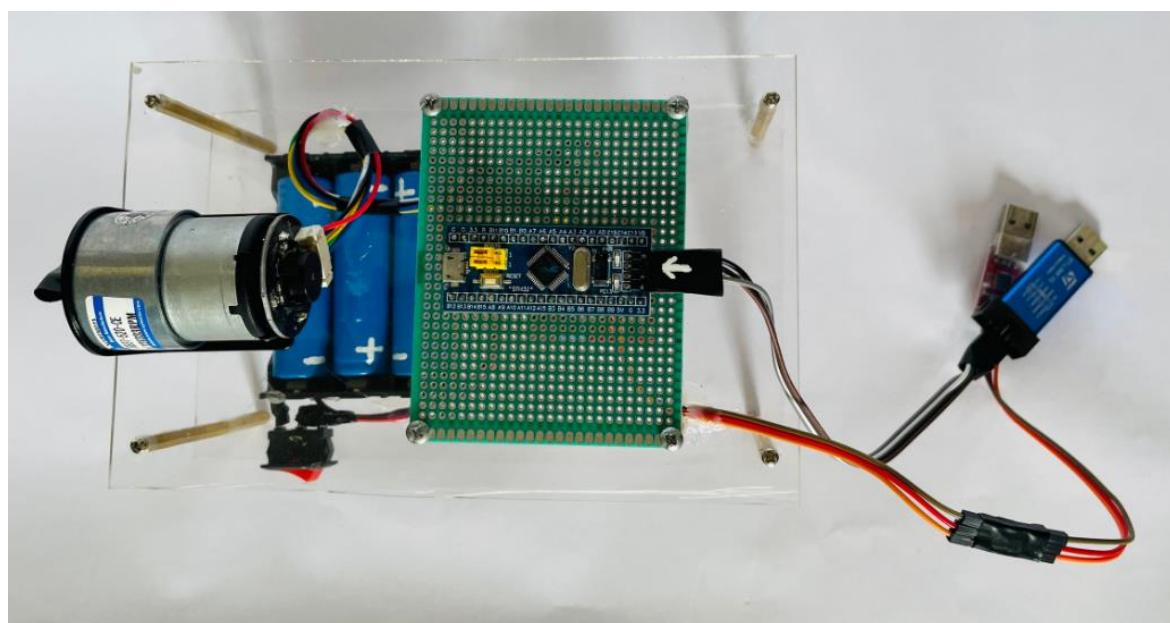


CHƯƠNG VI: ĐÁNH GIÁ CHẤT LƯỢNG HỆ THỐNG

Trong phần này, trên mô hình hệ thống đã hoàn chỉnh, ta sẽ khảo sát chất lượng đáp ứng ngõ ra khi thay đổi các thông số điều khiển. Từ đó rút ra phương pháp chỉnh thông số PID tốt nhất cho hệ thống.



Hình 17: Mô hình điều khiển động cơ DC hoàn chỉnh (mặt trước)

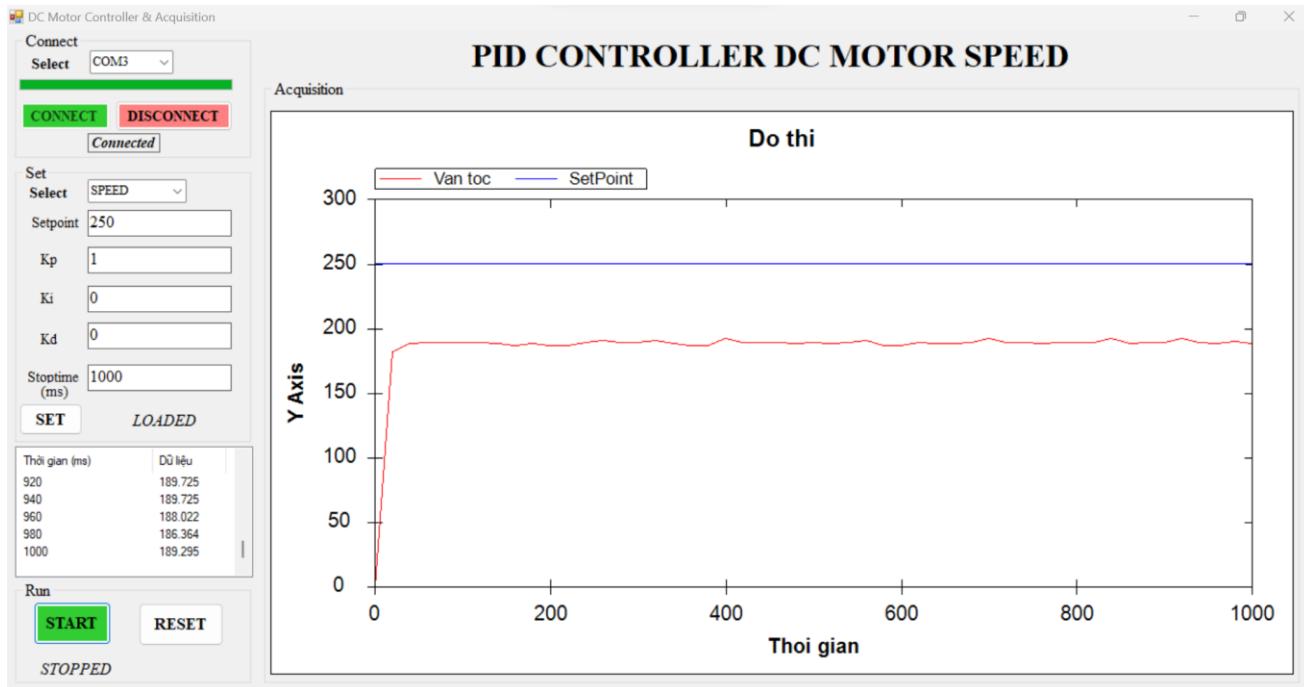


Hình 18: Mô hình điều khiển động cơ DC hoàn chỉnh (mặt trên)

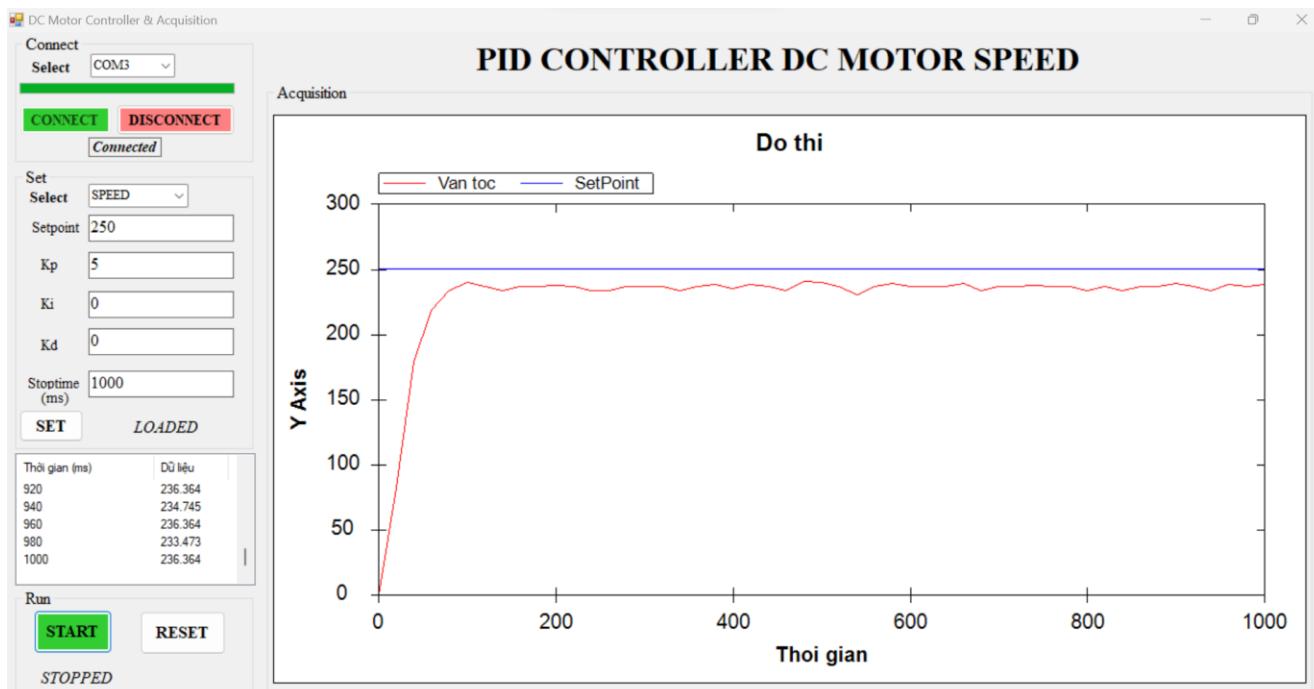
6.1. Điều khiển vận tốc

Vận tốc đặt 200 rpm.

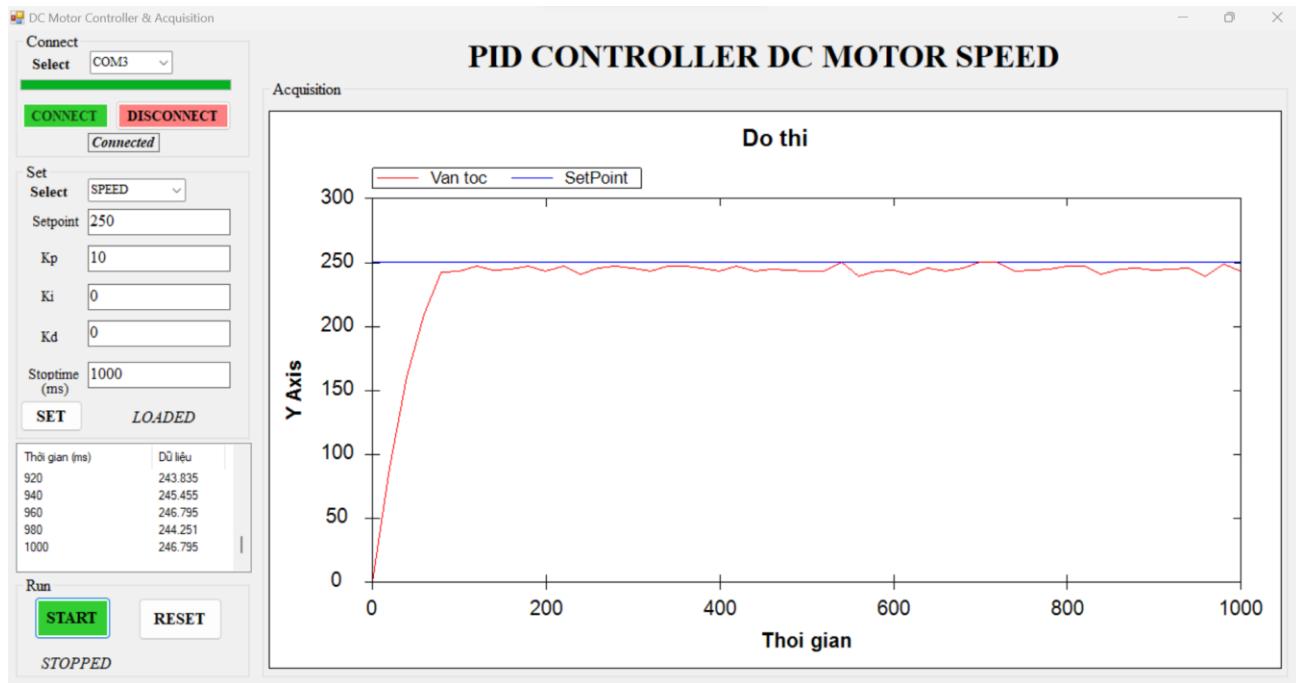
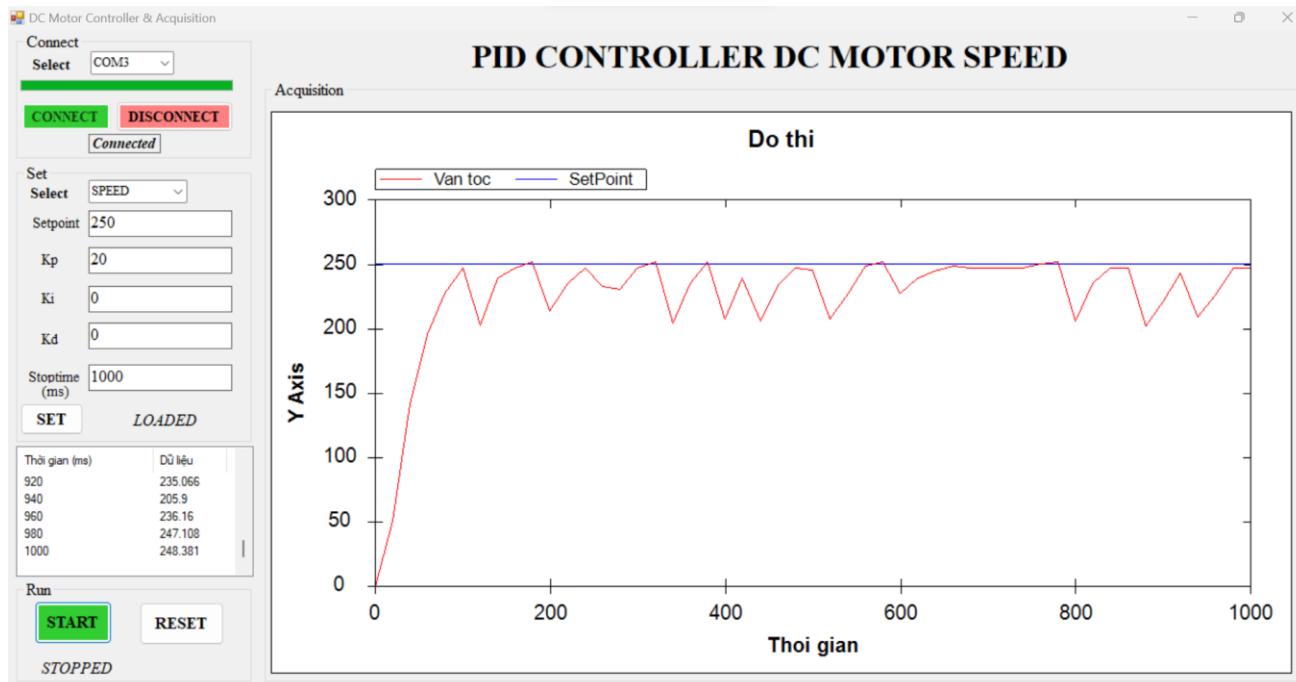
6.1.1. Thay đổi K_P (Bộ điều khiển P)



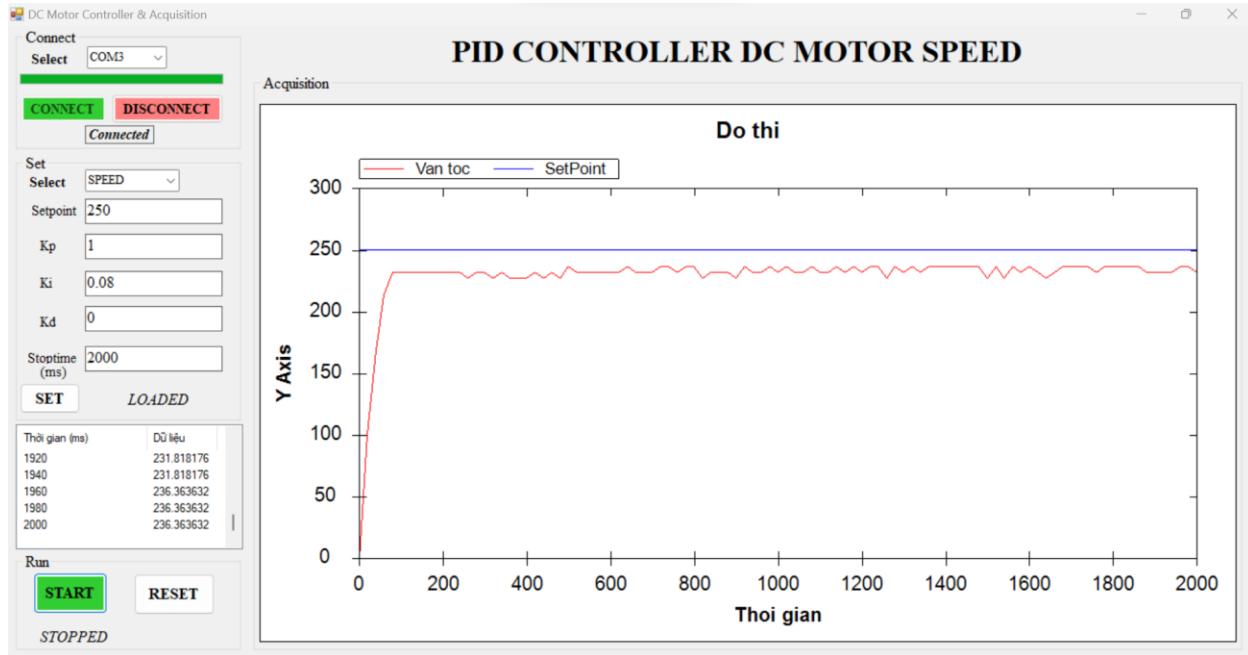
Hình 19: Đáp ứng vận tốc khi $K_P = 1, K_I = 0, K_D = 0$



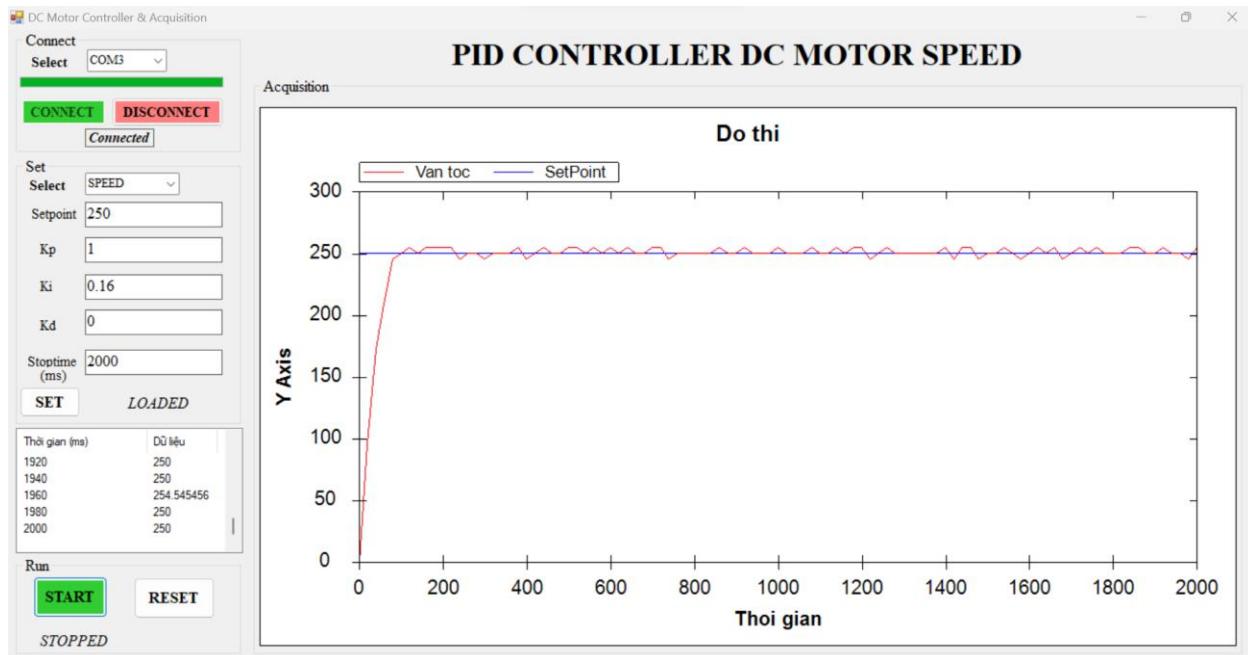
Hình 20: Đáp ứng vận tốc khi $K_P = 5, K_I = 0, K_D = 0$

Hình 21: Đáp ứng vận tốc khi $K_P = 10, K_I = 0, K_D = 0$ Hình 22: Đáp ứng vận tốc khi $K_P = 20, K_I = 0, K_D = 0$

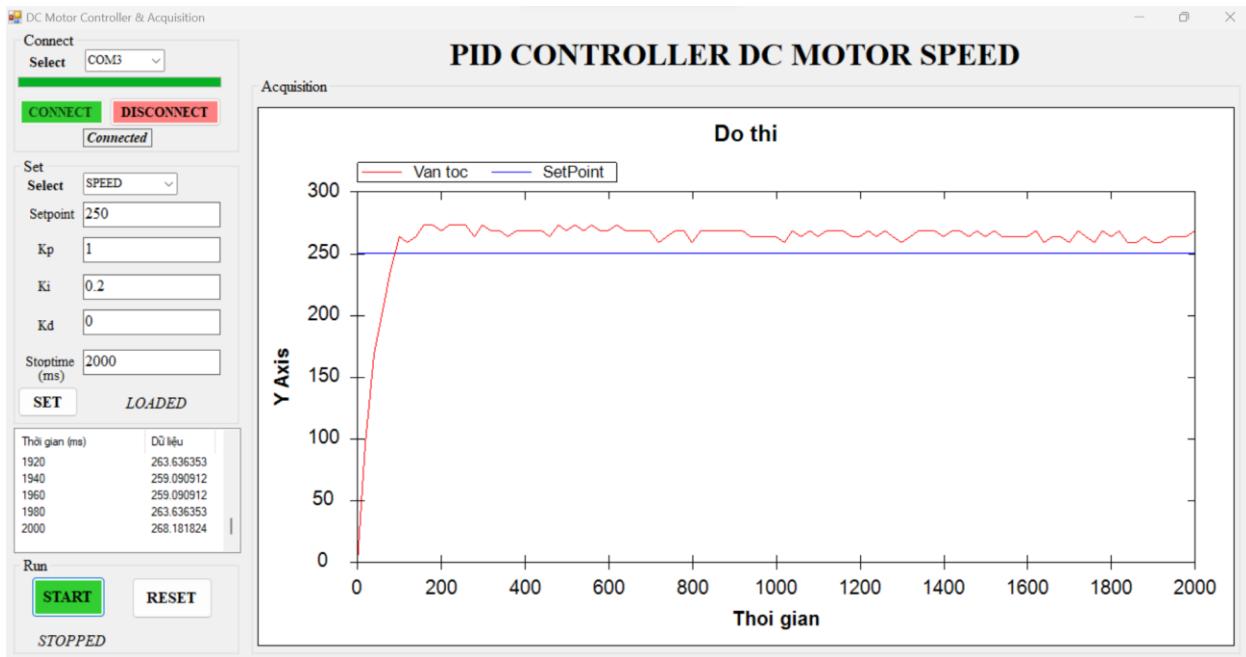
6.1.2. Thay đổi K_I (Bộ điều khiển PI)



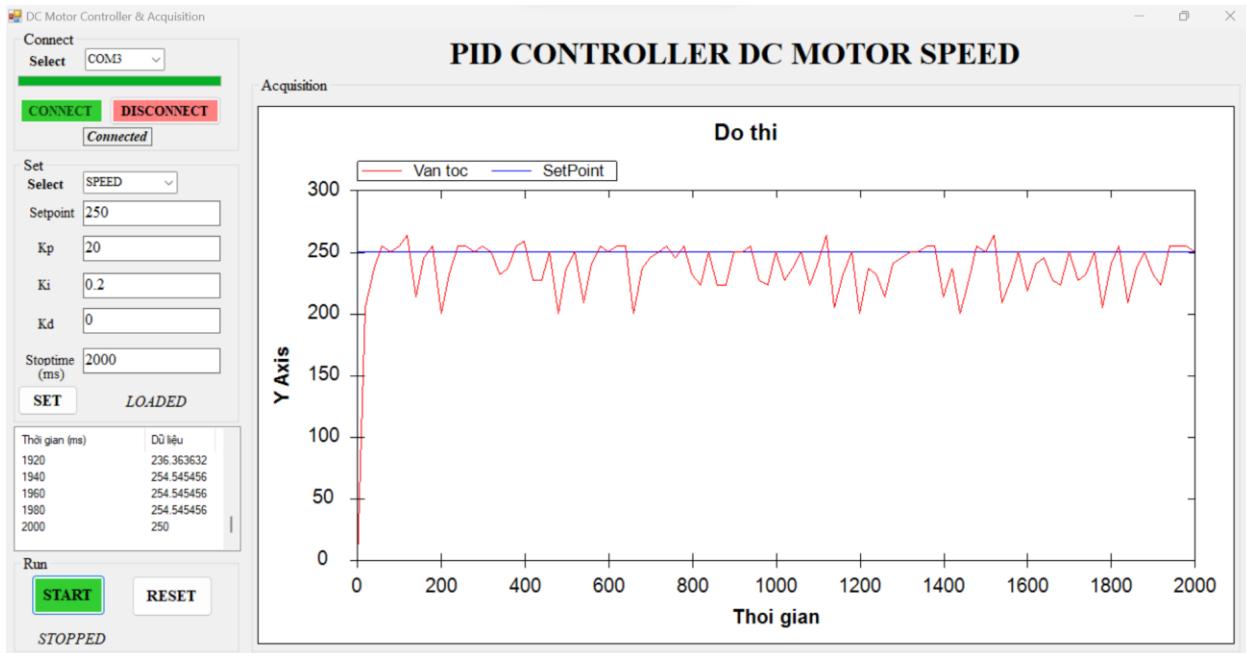
Hình 23: Dáp ứng vận tốc khi $K_P = 1$, $K_I = 0.08$, $K_D = 0$



Hình 24: Dáp ứng vận tốc khi $K_P = 1$, $K_I = 0.16$, $K_D = 0$

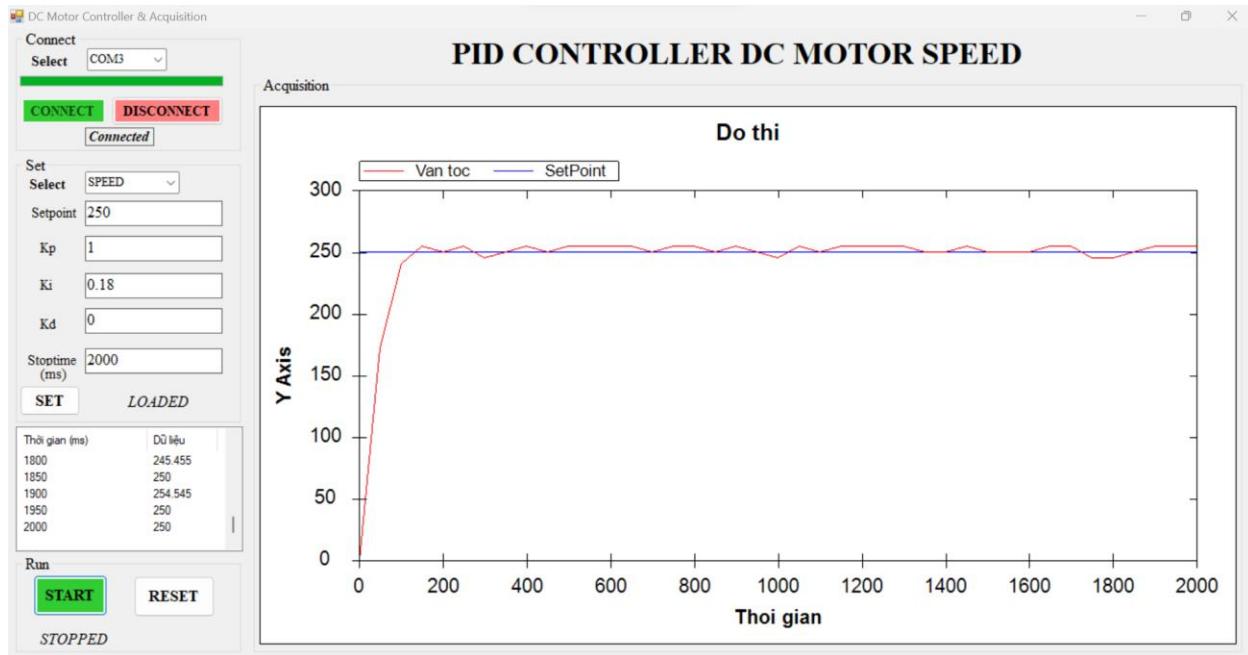


Hình 25: Dáp ứng vận tốc khi $K_P = 1$, $K_I = 0.2$, $K_D = 0$

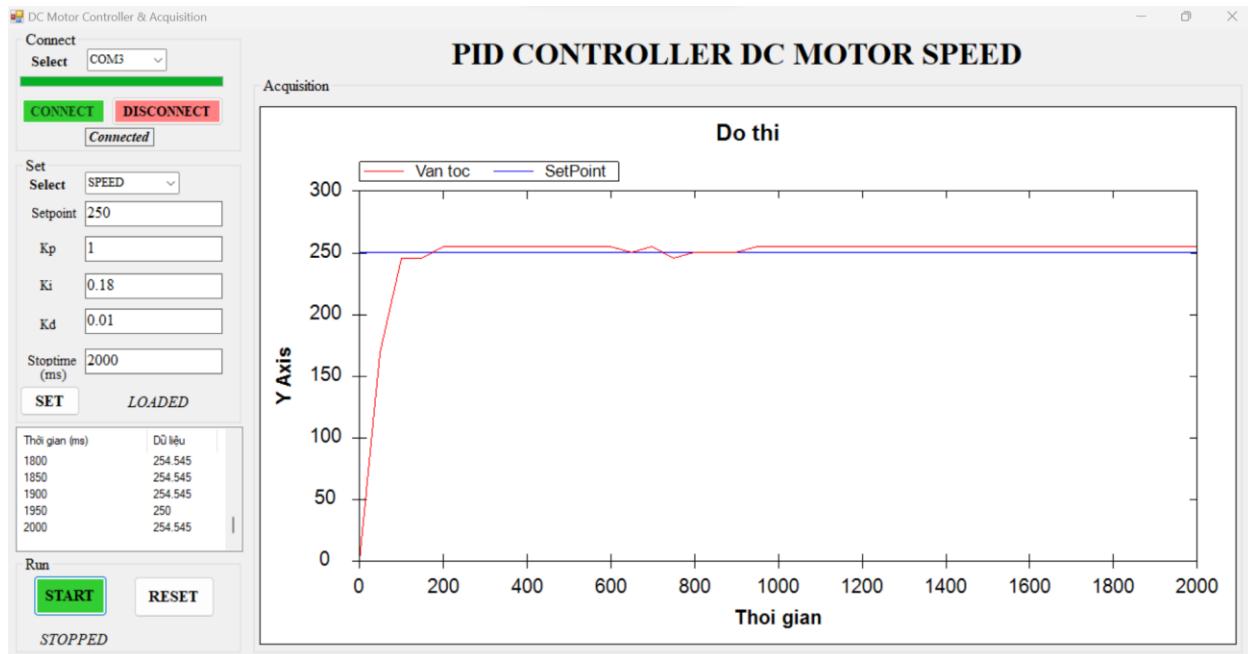


Hình 26: Dáp ứng vận tốc khi $K_P = 20$, $K_I = 0.2$, $K_D = 0$

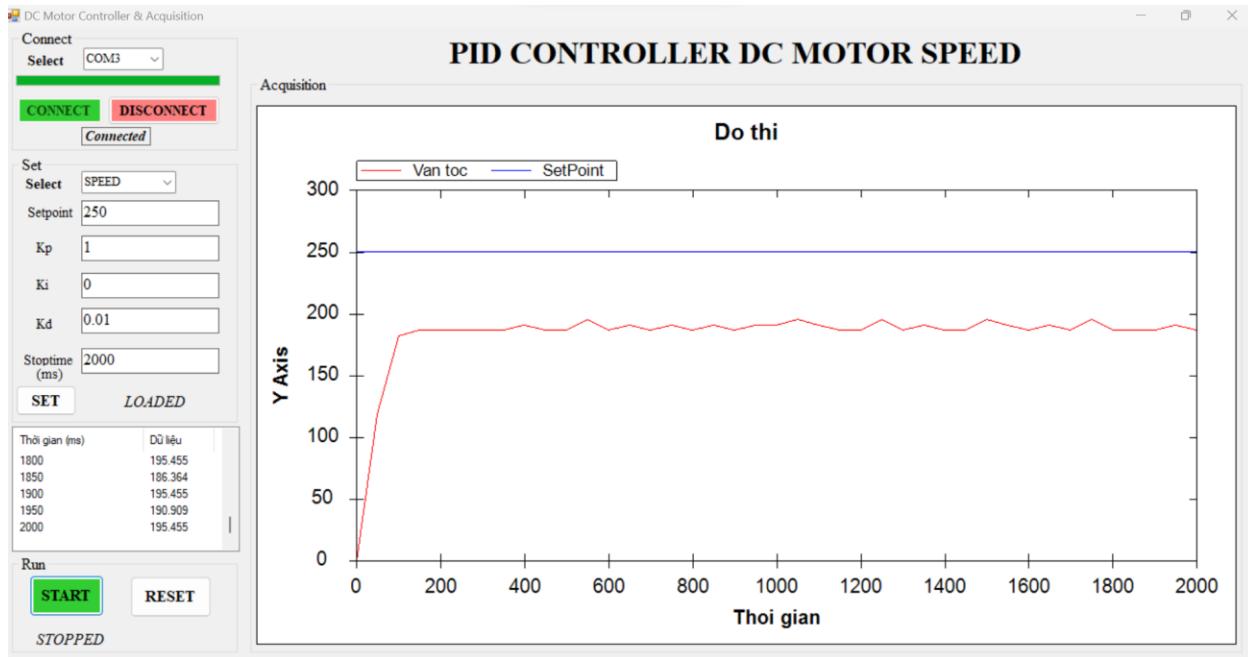
6.1.3. Thay đổi K_D (Bộ điều khiển PD và PID)



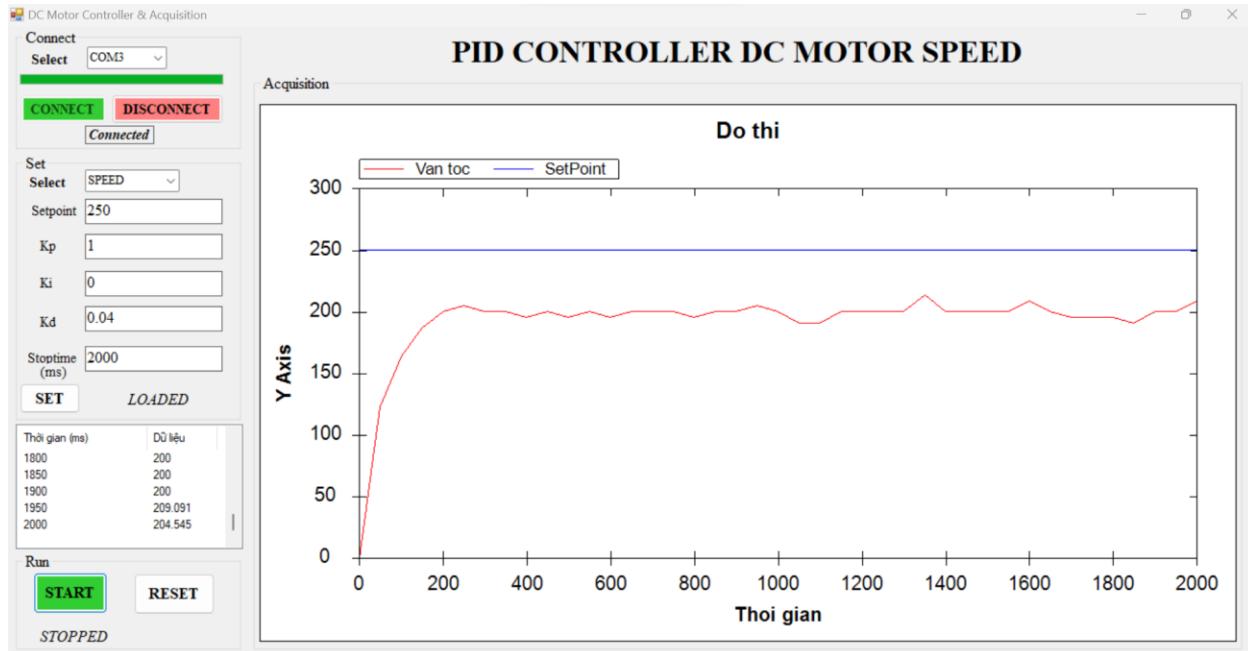
Hình 27: Dáp ứng vận tốc khi $K_P = 1$, $K_I = 0.18$, $K_D = 0$



Hình 28: Dáp ứng vận tốc khi $K_P = 1$, $K_I = 0.18$, $K_D = 0.01$



Hình 29: Đáp ứng vận tốc khi $K_P = 1$, $K_I = 0$, $K_D = 0.01$

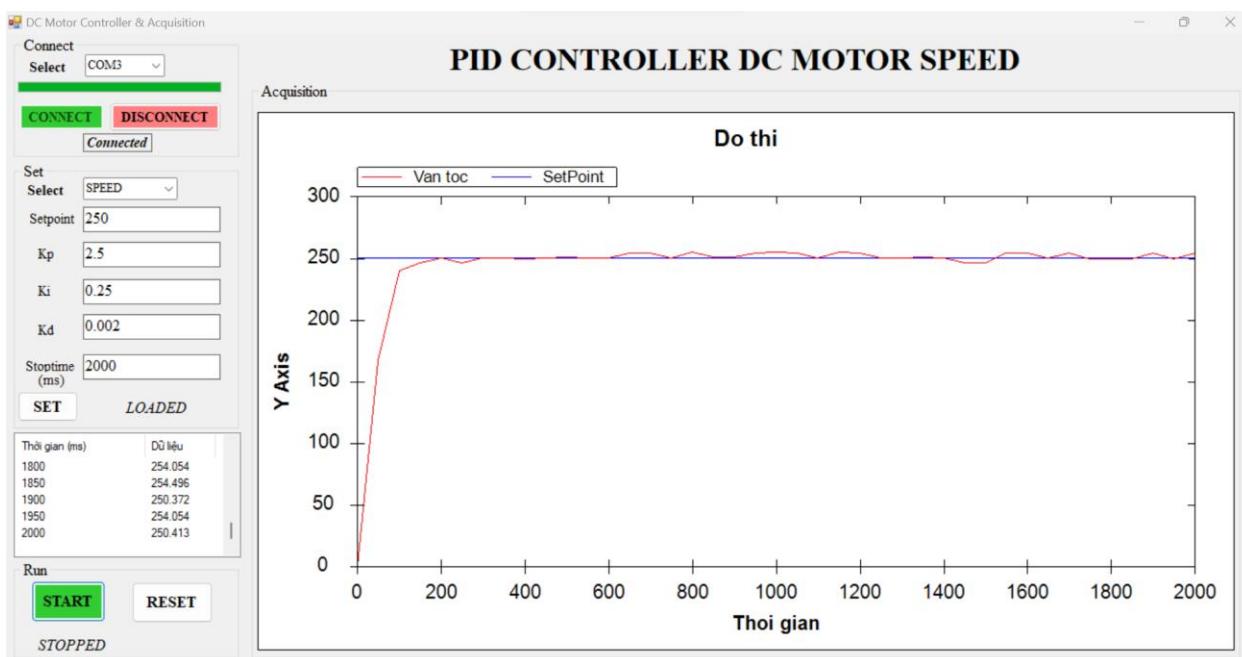


Hình 30: Đáp ứng vận tốc khi $K_P = 1$, $K_I = 0$, $K_D = 0.04$

6.1.4. Nhận xét

Từ kết quả chạy thí nghiệm trên, ta thấy:

- Kp tăng: thời gian quá độ và sai số xác lập giảm; hệ măt ổn định khi Kp quá lớn.
 - Ki tăng: sai số xác lập triệt tiêu; vọt lô tăng, thời gian quá độ tăng, đáp ứng dao động dài, hệ măt ổn định khi Ki quá lớn.
 - Kd tăng: vọt lô giảm; thời gian xác lập dài, đáp ứng chậm, sai số xác lập tăng.
- => Từ nhận xét trên, ta điều chỉnh thông số để có đáp ứng tốt nhất.

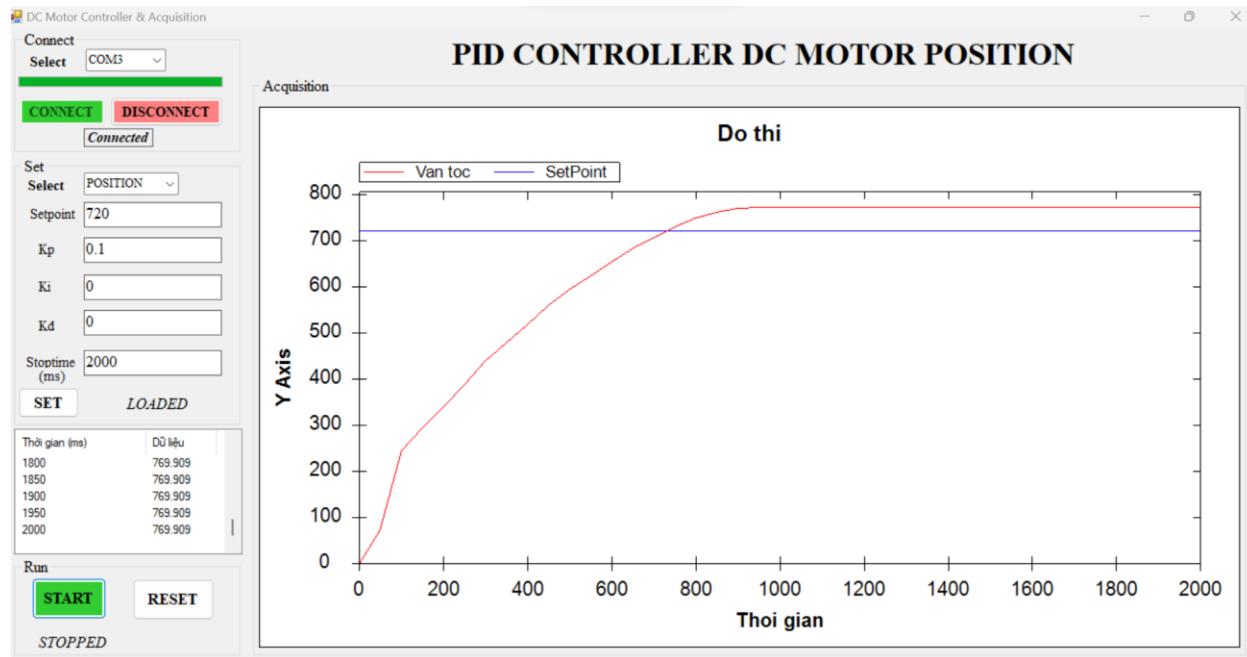


Hình 31: Đáp ứng vận tốc khi $K_p = 1$, $K_i = 0.25$, $K_d = 0.002$

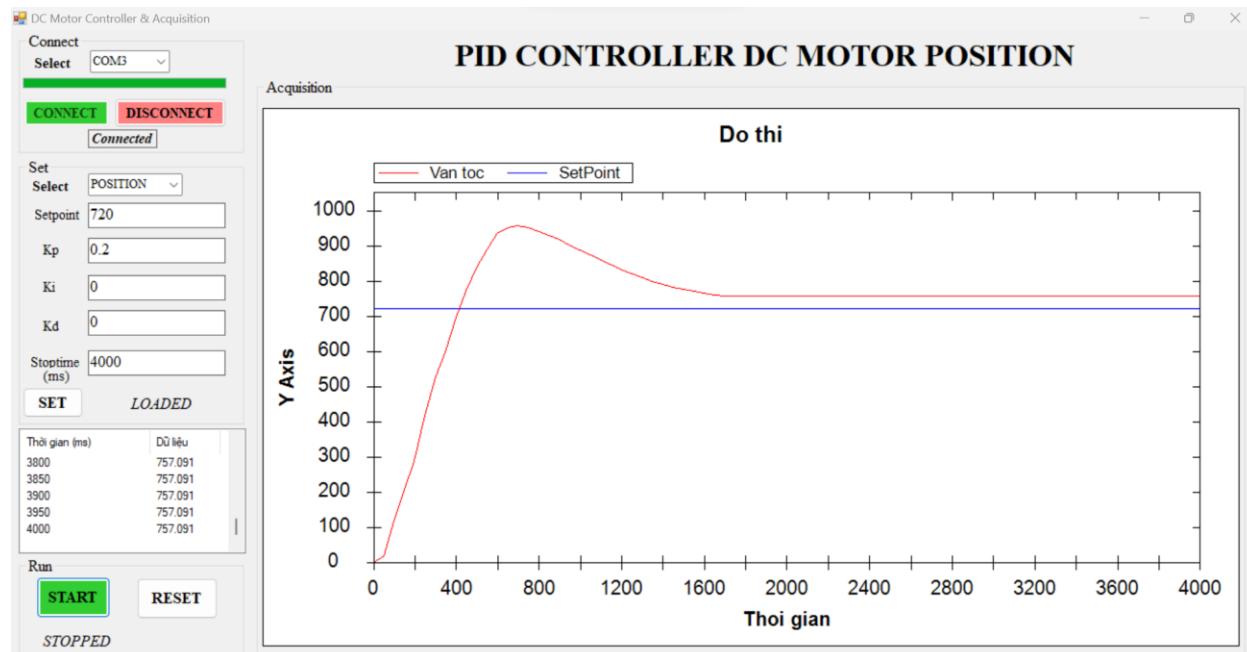
6.2. Điều khiển vị trí

Vị trí đặt là 720 độ.

6.2.1. Thay đổi K_P (Bộ điều khiển P)

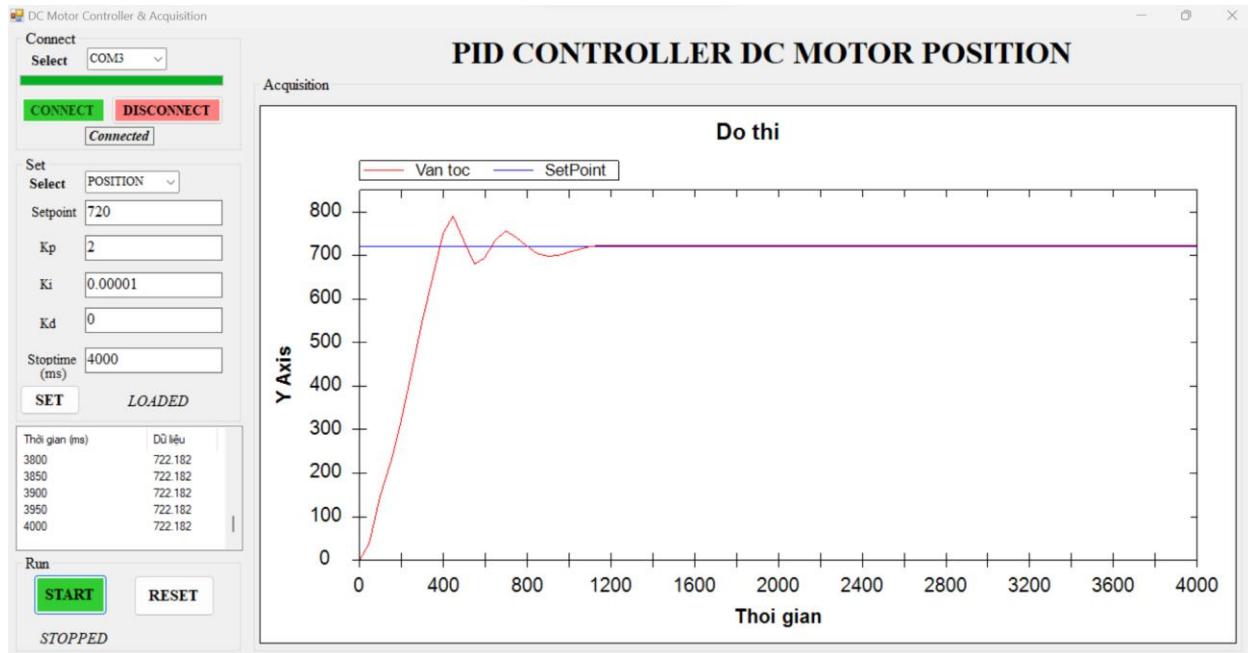


Hình 32: Dáp ứng vận tốc khi $K_P = 0.1$, $K_I = 0$, $K_D = 0$

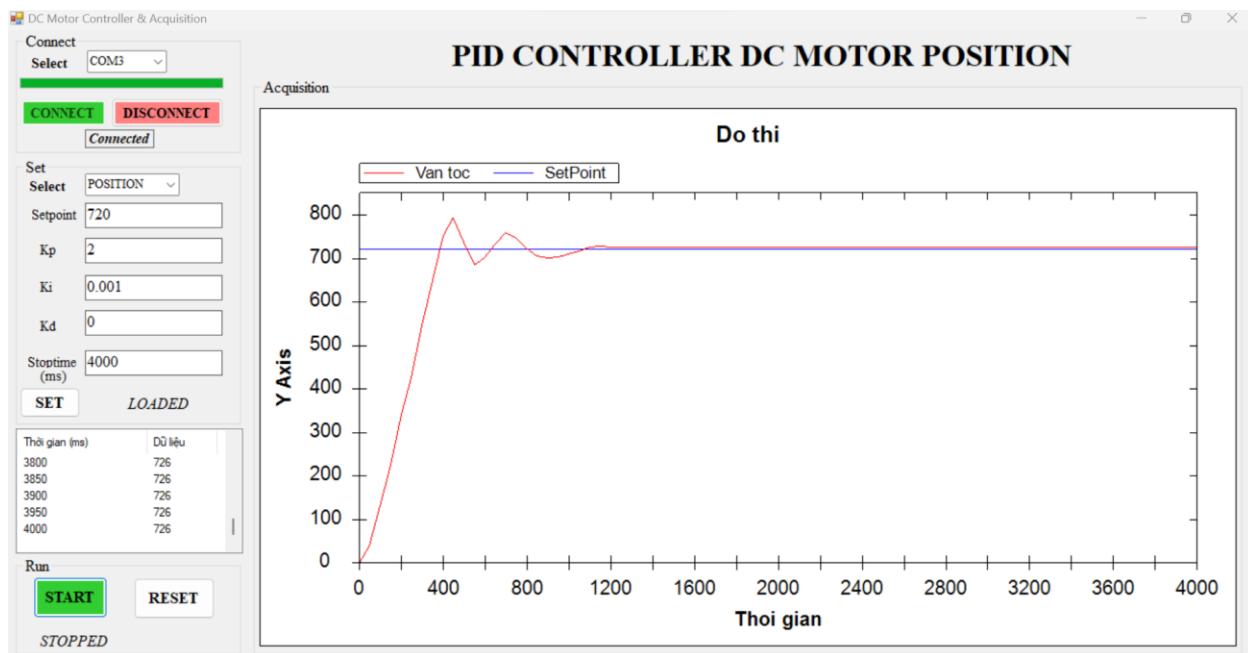


Hình 33: Dáp ứng vận tốc khi $K_P = 0.1$, $K_I = 0$, $K_D = 0$

6.2.2. Thay đổi K_I (Bộ điều khiển PI)

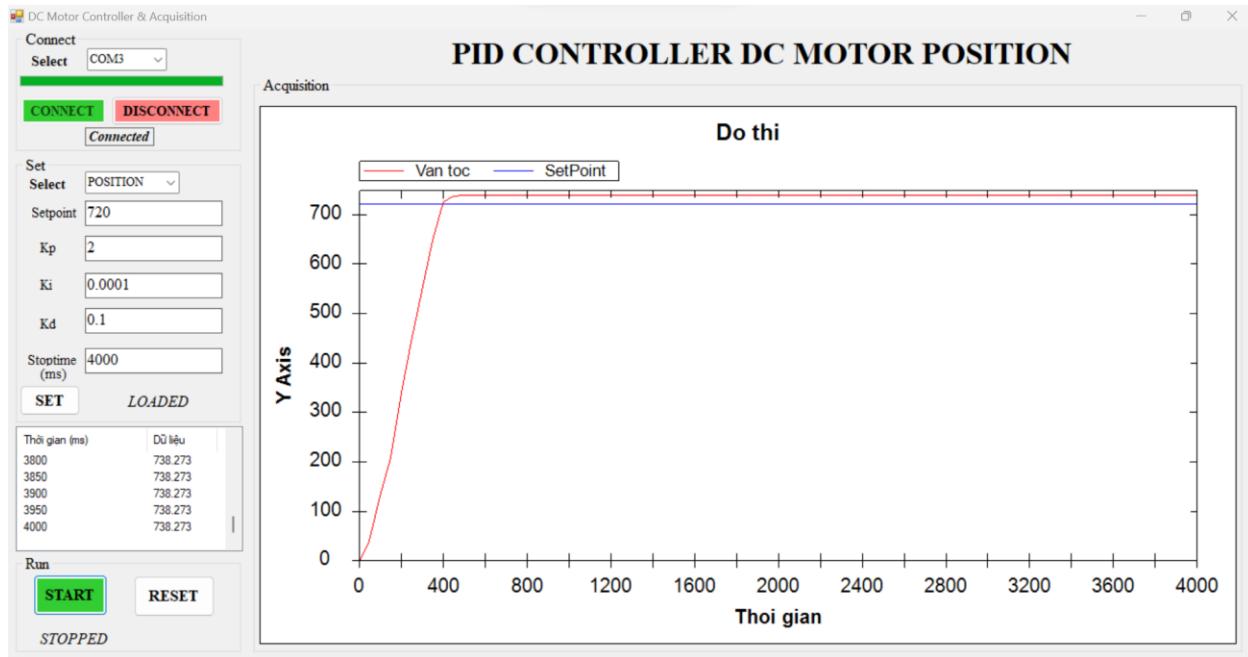


Hình 34: Đáp ứng vận tốc khi $K_P = 2$, $K_I = 0.00001$, $K_D = 0$

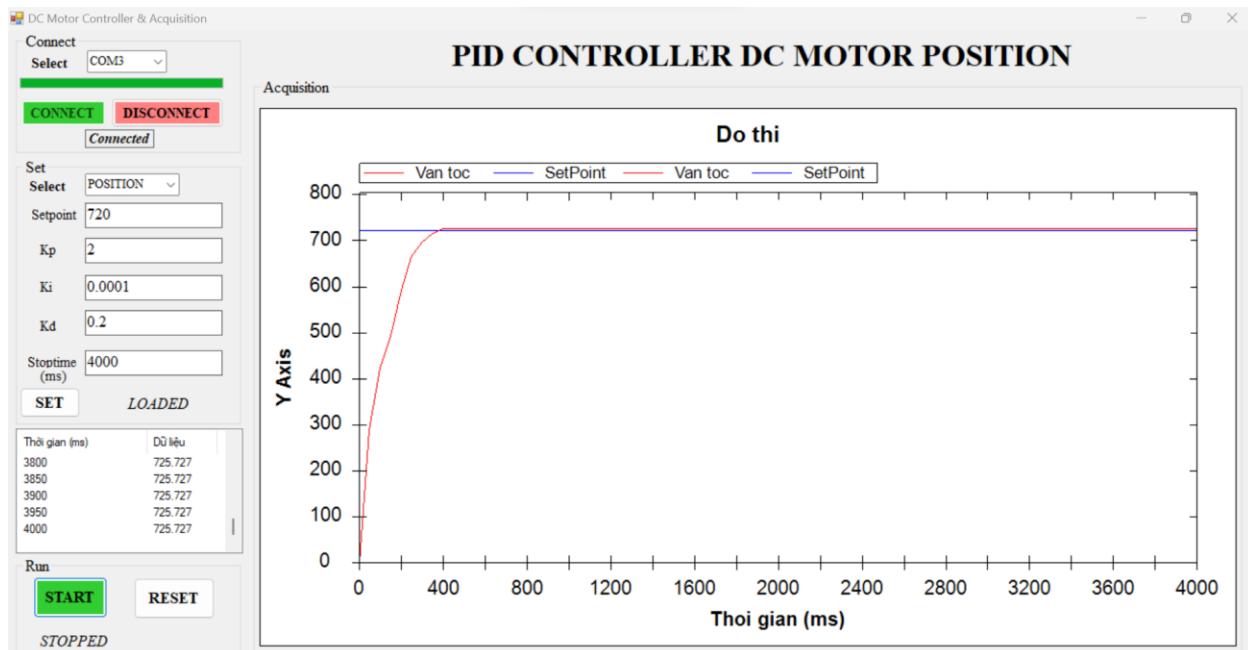


Hình 35: Đáp ứng vận tốc khi $K_P = 2$, $K_I = 0.001$, $K_D = 0$

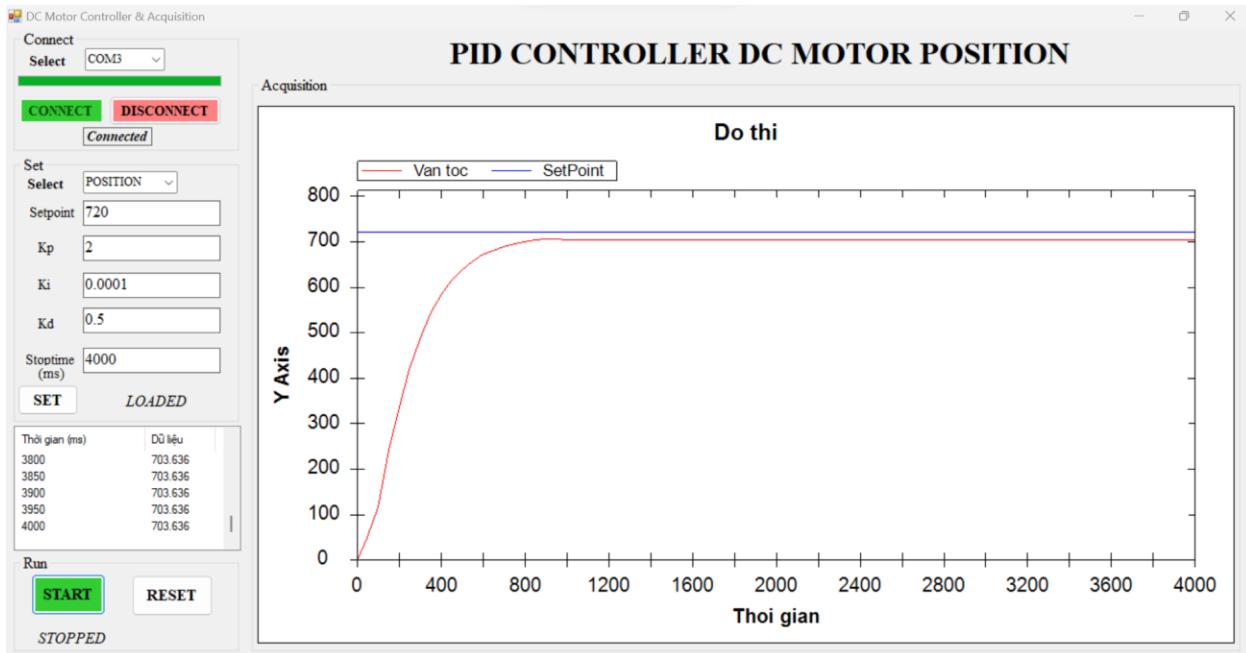
6.2.3. Thay đổi Kd (Bộ điều khiển PID)



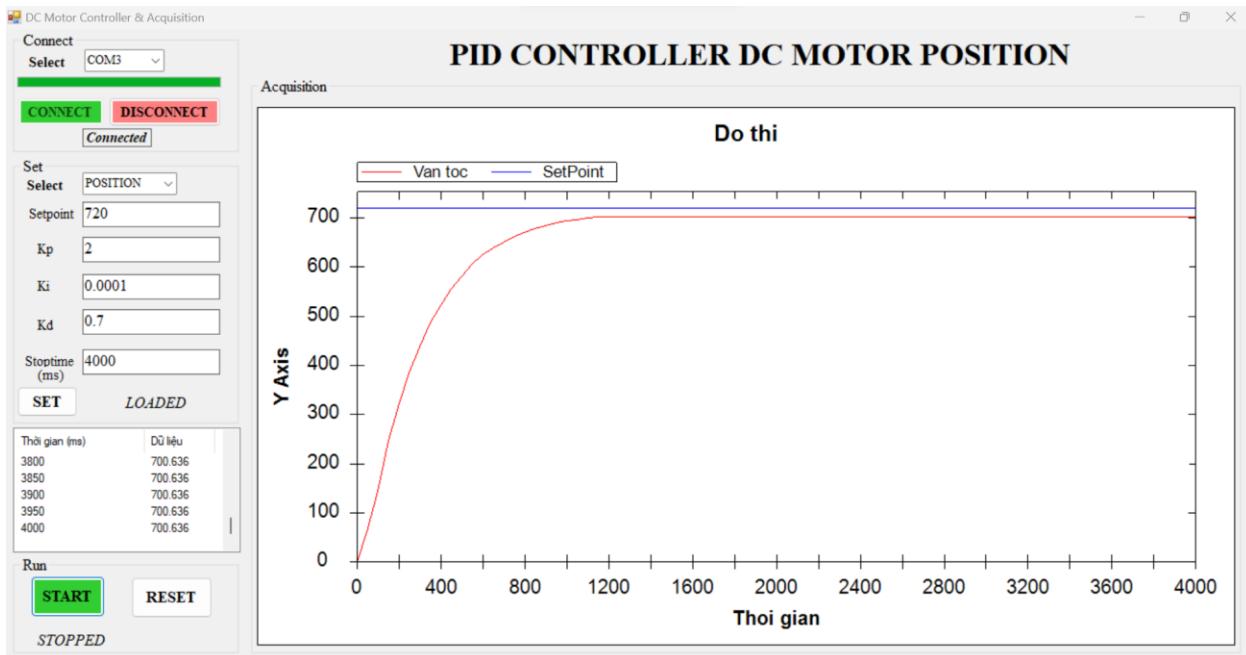
Hình 36: Dáp ứng vận tốc khi $K_P = 2$, $K_I = 0.0001$, $K_D = 0.1$



Hình 37: Dáp ứng vận tốc khi $K_P = 2$, $K_I = 0.0001$, $K_D = 0.2$



Hình 38: Đáp ứng vận tốc khi $K_P = 2$, $K_I = 0.0001$, $K_D = 0.5$

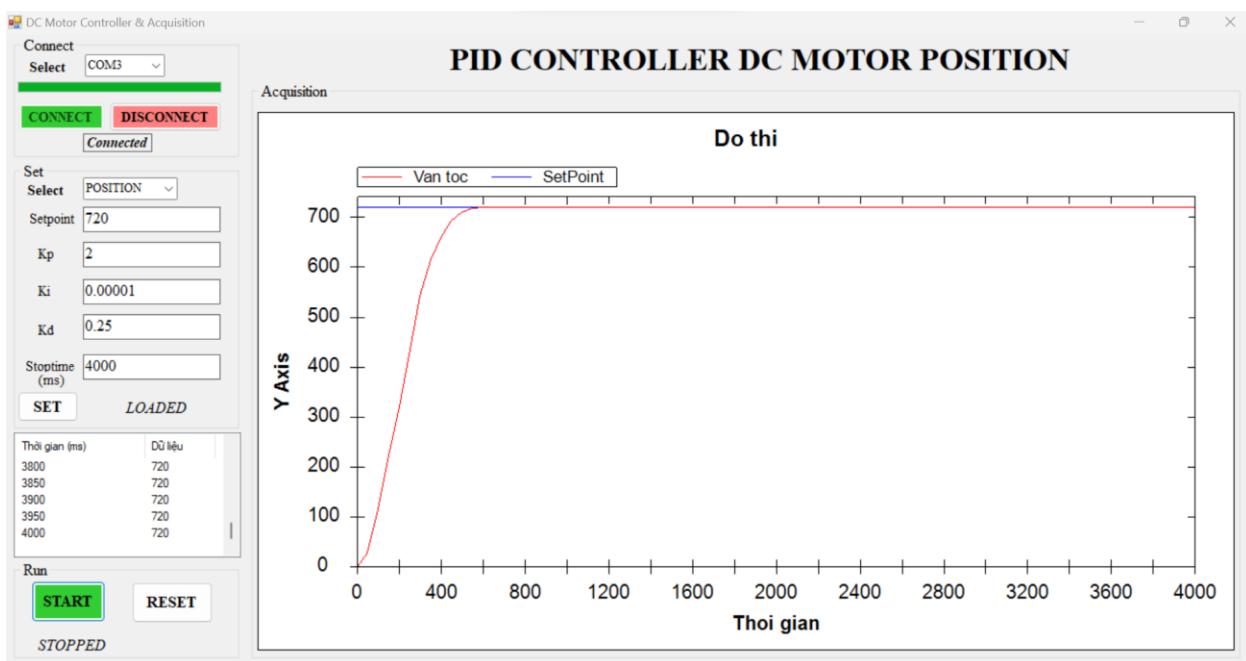


Hình 39: Đáp ứng vận tốc khi $K_P = 2$, $K_I = 0.0001$, $K_D = 0.7$

6.2.4. Nhận xét

Từ kết quả chạy thí nghiệm trên, ta thấy:

- Kp tăng: thời gian quá độ và sai số xác lập giảm; hệ măt ổn định khi Kp quá lớn.
 - Ki tăng: sai số xác lập triệt tiêu; vọt lô tăng, thời gian quá độ tăng, đáp ứng dao động dài, hệ măt ổn định khi Ki quá lớn.
 - Kd tăng: vọt lô giảm; thời gian xác lập dài, đáp ứng chậm, sai số xác lập tăng.
- => Từ nhận xét trên, ta điều chỉnh thông số để có đáp ứng tốt nhất.



Hình 40: Đáp ứng vận tốc khi $K_p = 2$, $K_i = 0.00001$, $K_d = 0.25$

CHƯƠNG VII: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

7.1. Kết luận

Sau khi thực hiện đề tài, ta rút ra các kết luận sau

- Về bộ điều khiển: bộ điều khiển PID phù hợp với đối tượng động cơ DC, với thông số phù hợp có thể cho đáp ứng nhanh, vọt lố nhỏ và sai số ít. Qua khảo sát, hiểu được quy luật ảnh hưởng của mỗi thông số lên hệ thống, từ đó có dự báo và điều chỉnh tốt. Nhược điểm của bộ điều khiển PID là đáp ứng xấu khi đối tượng thay đổi do các yếu tố như động cơ mang tải, cuộn dây nóng lên, chổi than mòn, môi trường từ tính...
- Về phần cứng: mạch lái thực hiện được chức năng cung cấp công suất cho động cơ tuy nhiên có hạn chế khi cung cấp dòng lớn để động cơ quay ở tốc độ cao hay mang tải. Opto thực hiện cách ly hiệu quả vi điều khiển khỏi dòng áp công suất cao. Tuy nhiên việc opto không nằm trong mạch lái làm phần cứng chiếm nhiều diện tích.
- Về phần nhúng: kit STM32F103C8T6 có hiệu năng cao, tốc độ xử lý nhanh, đáp ứng được cùng lúc các yêu cầu tính toán, phát xung, đọc cảm biến, giao tiếp máy tính.
- Về phần mềm: công cụ Visual Studio giúp máy tính thực hiện chức năng giao tiếp UART với vi điều khiển nhanh và chính xác. Tuy nhiên nó có ít thư viện chức năng, giao diện đồ họa hạn chế, hệ ngôn ngữ khá lỗi thời, khiến cho việc lập trình mất nhiều thời gian.

Một số phương án cải tiến cho đề tài:

- Sử dụng các bộ điều khiển có tính thích nghi như MRAS, STR, logic mờ... để hệ vẫn có chất lượng tốt khi đối tượng thay đổi.
- Thiết kế mạch lái có công suất cao hợp, tích hợp sẵn opto, đồng thời có chốt ra chân cho vi điều khiển để tạo thành một module duy nhất.
- Sử dụng các vi điều khiển khác nhưng vẫn đầy đủ chức năng như STM32F407, PIC16F877, AT89C52...
- Sử dụng các công cụ lập trình giao diện hiệu quả hơn như LabVIEW, Java, Qt designer...

Tùy theo mục đích điều khiển và những yêu cầu về tối ưu để ta có thể xác định được việc đặt những thông số PID thích hợp nhất. Cách tìm được các thông số này có thể sử dụng các phương pháp tính toán từ thực nghiệm như:

- + Điều chỉnh thủ công bằng thực nghiệm. Phương pháp này đơn giản nhất nhưng đòi hỏi người kỹ sư phải có chuyên môn cao, ước tính gần đúng nhất các thông số thích hợp thì quá trình tìm ra thông số tốt nhất sẽ nhanh chóng.
- + Phương pháp cổ điển nhất được áp dụng khá nhiều đó là phương pháp Ziegler– Nichols thường được ứng dụng nhiều và đưa vào chương trình giảng dạy. Phương pháp này sử dụng khá đơn giản vì sẽ chạy thực nghiệm, không cần tính toán nhiều, tuy nhiên sẽ rất mất thời gian thử nhiều lần, nếu thử những thông số quá xấu có thể khiến hiện tượng dao động rất mạnh.
- + Khi quá trình phát triển của công nghệ kỹ thuật ngày càng mạnh mẽ, và ứng dụng AI ngày càng được ứng dụng rộng rãi thì các phần mềm tự điều chỉnh thông số được ra đời. Các phần mềm hỗ trợ rất nhiều trong việc tìm chính xác và nhanh chóng các hệ số PID phù hợp, tuy nhiên giá cả khá cao và cần phải được huấn luyện nhiều mới có thể áp dụng tốt vào thực tiễn.

Hiện tại, công thức tính toán PID được sử dụng ở đây là công thức cổ điển sơ khai nhất của bộ điều khiển PID số, nên khả năng điều khiển sẽ còn nhiều mặt hạn chế, không thể đạt được kết quả cao. Vì thế, thuật toán PID ngày càng phát triển tạo ra những bản thể khác để phục vụ tốt hơn thiết kế chuyên môn hóa hơn cho các đối tượng điều khiển khác nhau. Những bộ PID phát triển khác có thể kể đến như:

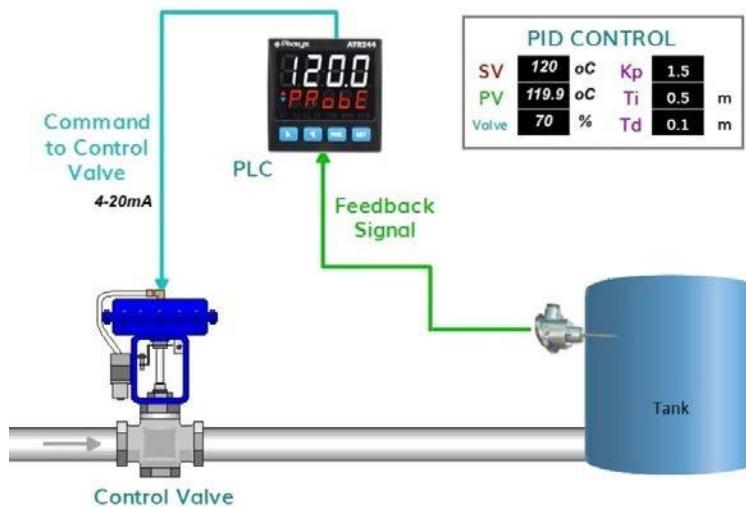
- + *Tác động vào khâu tích phân*: Các thực hiện thay vì để so sánh sai số với Setpoint, ta có thể mở rộng so sánh sai với $Setpoint \pm \epsilon * Setpoint$. Trong số ϵ sẽ xác định vùng làm việc của khâu tích phân. Nếu giá trị hiện tại của hệ thống nằm trong khoảng $Setpoint \pm \epsilon * Setpoint$ thì sẽ ngắt khâu I ra khỏi bộ điều khiển, chỉ sử dụng PD. Vì khâu tích phân có thành phần cộng dồn sai số nên làm vọt lô và duy trì liên tục việc tăng sai số tích lũy bị gián đoạn. Khi giá trị hiện tại nằm ngoài phạm vi này ta sẽ thêm khâu I lại hệ thống.

+ *Dốc hóa điểm giá trị đặt.* Phương pháp sẽ không để Setpoint vào ngay khi vừa mới bắt đầu chạy, mà sẽ thay đổi thành một phương trình đường thẳng có độ dốc theo một hệ số góc k chọn trước, từ đó sẽ làm cho sai số tính toán trong bộ điều khiển luôn nhỏ, thì hệ thống sẽ làm việc mượt hơn, ổn định hơn, không bị dao động nhiều, tránh những tác động đột ngột. Phương pháp này hiệu quả với các hoạt động của hệ thống thay đổi giá trị đặt liên tục trong quá trình chạy.

7.2. Hướng phát triển

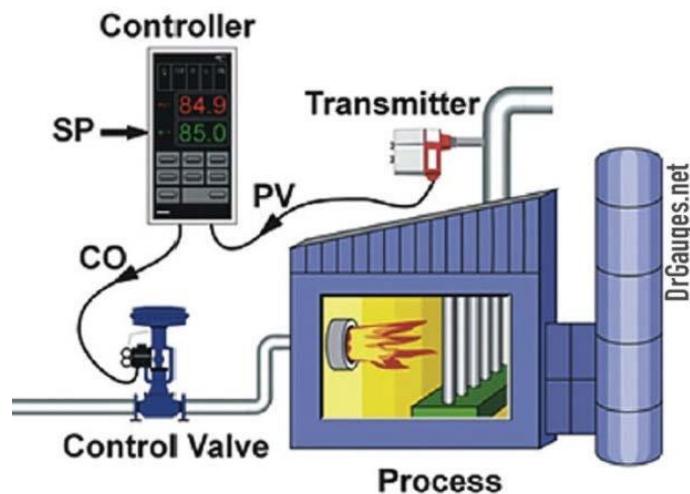
Một trong những ứng dụng gần gũi nhất đối với sinh viên ngành tự động trong việc sử dụng bộ điều khiển PID điều khiển động cơ vào trong việc làm xe cân bằng. Để cân bằng một chiếc xe 2 bánh mang phía trên một tải có trọng lượng lớn sẽ cần đến bộ điều khiển PID giúp xe xác định được góc nghiêng hiện tại của xe so với góc thẳng đứng để thay đổi chiều quay của động cơ giúp xe luôn được giữ vững.

Ứng dụng tiếp đến đó là đóng mở van bơm nước hoặc chất lỏng khác với nhiệt lượng phù hợp vào bồn chứa. Nếu như ở thời điểm khoa học chưa phát triển, các nhà máy sẽ phải thuê các nhân công thường xuyên kiểm tra các bồn chứa để đảm bảo lượng nước hoặc chất lỏng cần dùng không được hết trong quá trình sản xuất và nhiệt độ phải được đo thường xuyên để đảm bảo đúng nhiệt độ, rất tốn kém và gặp nhiều khó khăn. Thì ngày nay với sự phát triển của nhiều loại cảm biến kết hợp với bộ điều khiển PID trong bộ hệ thống xử lý PLC sẽ có thể thay thế con người trong việc giám sát và mở van nước bơm vào bồn chứa một cách nhanh chóng và hiệu quả nhất.



Hình 41: Bộ điều khiển PID trong việc điều khiển bơm nước vào bồn chứa với nhiệt độ thích hợp

Có thể kể đến việc tích hợp bộ điều khiển PID vào lò nung nhiệt độ. Như vậy việc bật tắt lò để điều chỉnh công suất lò cấp nhiệt sẽ không còn do người điều khiển nữa mà sẽ do hệ thống điều khiển tự tác động vào bằng việc đo đạt giá trị nhiệt độ hiện tại để đưa vào bộ điều khiển tính toán xử lí và đưa ra mức công suất phù hợp để tăng nhiệt, giữ nhiệt, hoặc tắt lò để giảm nhiệt một cách hiệu quả nhất.



Hình 42: Ứng dụng PID trong điều khiển nhiệt độ của lò nung vật liệu