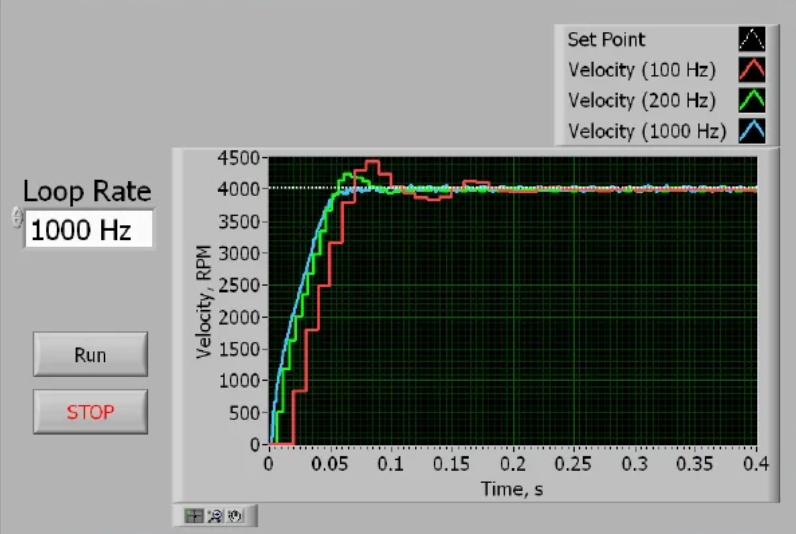
**CẢI THIỆN CHẤT LƯỢNG PID**

1. **Cải thiện phần cứng - Tăng tần số của vòng lặp điều khiển :**



Dựa vào băng thông của hệ thống cần điều khiển, định lý Shannon – Nyquist và giới hạn của phần cứng hệ thống ta sẽ xác định được điều kiện chọn tần số điều khiển, từ đó ta chọn tần số của vòng lặp điều khiển lớn nhất có thể để tối ưu chất lượng đáp ứng.

Trong phương pháp này băng thông của hệ thống sẽ quyết định cận dưới của điều kiện chọn tần số và cận trên của điều kiện này quyết định bới phần cứng hệ thống, mà để hệ thống có đáp ứng tốt hơn thì thông thường ta phải tăng tần số điều khiển hay nói cách khác ta phải cải thiện phần cứng hệ thống dẫn đến giá thành thiết kế tăng

1. **Cải thiện chất lượng tín hiệu ngõ vào bộ điều khiển PID :**



Trước tiên chúng ta hãy cùng nhìn công thức của khâu D trong bộ PID, như đã thảo luận ở trên thì khi ta cố gắn tăng tần số điều khiển hay nói cách khác là giảm thời gian lấy mẫu thì đồng nghĩa rằng Dt sẽ rất nhỏ, điều này dẫn đến việc khâu vi phân sẽ rất nhạy với các nhiễu của tín hiệu sai số hay nói cách khác là tín hiệu trả về của cảm biến, bên cạnh đó khâu P và khâu I cũng sẽ bị ảnh hưởng nếu việc đọc cảm biến có nhiễu môi trường hoặc xuất hiện sai số do việc lượng tử hóa các giá trị thực thì sai số hay nhiễu ấy sẽ ảnh hưởng rất lớn đối với chất lượng điều khiển.

Giải pháp là:

Sử dụng sử dụng các thiết bị có sai số lượng tử hóa nhỏ, sử dụng vùng dữ liệu phù hợp và ưu tiên thực hiện phép tính nhân trước khi thực hiện các phép tính chia trong các biểu thức chuyển đổi để giảm thiểu các sai số lượng tử hóa do cơ cấu làm tròn số thực của phần cứng ,

Sử dụngcác phương pháp chống nhiễu vật lý cho đường dây dẫn, các phương pháp ước lượng hoặc lọc nhiễu tín hiệu cảm biến như: Lọc thông thấp, lọc bù, lọc trung vị; lọc trung bình; lọc Kalman;…

1. **Cải thiện chất lượng ngõ ra của bộ điều khiển PID :**

Ngõ ra của bộ điều khiển PID theo thuật toán là không có giới hạn cụ thể nào tuy nhiên giá trị này bị chặn bởi giá trị của phần cứng khâu công suất chấp hành, ta phải giới hạn ngõ ra bộ PID nằm trong vùng an toàn của khâu công suất chấp hành.

Một điều quan trọng khác mà chúng ta cần quan tâm là luôn có những yếu tố môi trường tác động lên khâu công suất chấp hành mà ta cần phải offset để điều chỉnh.

Cho một ví dụ như sau: Đối với động cơ DC ở trong tình trạng không tải thì moment mà động cơ này sinh ra phải thắng được ma sát nghỉ của Rotor động cơ thì động cơ mới bắt đầu quay được, đối với động cơ lúc có tải thì moment mà động cơ sinh ra phải lớn hơn tổng của ma sát nghỉ và moment nghỉ trên đầu trục động cơ, lúc này sẽ tồn tại một dãy các giá ngõ ra của bộ PID chưa đủ lớn để thắng được ma sát nghỉ và moment nghỉ này ta tạm gọi là dãy giá trị chết. Đây là dãy giá trị mà tại đó giá trị ngõ ra của bộ PID thay đổi nhưng đáp ứng ngõ ra không đổi, do quá trình thiết kế bộ PID trên lý thuyết chưa bao gồm các vấn đề trên như một biến số của hệ thống.

Giải pháp là ta tiến hành offset ngõ ra bộ PID để loại bỏ dãy chết này tùy thuộc vào nguyên nhân chính sinh ra dãy giá trị chết này mà ta có thể xác định cách offset.

Đối với động cơ không tải: giá trị offset là một hằng số vừa đủ giúp động cơ thắng được ma sát nghỉ của chính nó giá trị offset này tạm gọi là A

Đối với động cơ có tải: giá trị offset lúc này sẽ phải bao gồm giá trị A vừa đề cập ở trên và một giá trị B, giá trị B này có một ứng suất thuận với độ lớn của moment tải trên đầu trục động cơ.

Trong một số hệ thống cân bằng ta có thể thêm một hệ số anpha có giá trị từ [0 1] để tinh chỉnh thành phần offset này giúp hệ thống dễ cân bằng hơn và tránh tình trạng đáp ứng bị “Cattering”.

1. **Sử dụng công thức PID phù hợp với tần số điều khiển :**

Trường hợp 1: Tần số điều khiển chưa đủ thỏa mãn các tính toán Shanon -> PID rời rạc

Trường hợp 2: Tần số điều khiển đã đủ thỏa mãn các tính toán Shanon -> [PID rời rạc hoặc PID liên tục]

1. **Trong quá trình rời rạc hóa sử dụng các phương pháp cải tiến các phép tính tích phân và vi phân ở miền rời rạc :**

………

1. **Sử dụng nhiều bộ thông số PID khác nhau cho các vùng hoạt động PID khác nhau:**

Do một bộ thông số PID cụ thể chỉ có thể hoạt động tốt trong lân cận của một Setpoint xác định cho nên khi Setpoint thay đổi nhiều và ra khỏi lân cận ổn định của bộ thông số PID này thì ta phải chuyển sang bộ PID khác mà tại đó bộ PID mới này hoạt động tốt trong lân cận của Setpoint mới, với phương pháp này ta sẽ có được vùng điều khiển của bộ PID rộng hơn, nhược điểm của phương pháp làm này là sẽ không đảm bảo tính ổn định của đáp ứng nếu bộ chuyển mạch giữa các bộ số PID không được khống chế chính xác.

1. **Sử dụng quy hoạch quỹ đạo :**

Sử dụng các phương pháp quy hoạch quỹ đạo như quintic, cubic, LSPB để kiểm soát các quá trình quá độ từ setpoint này đến setpoint khác, phương pháp này sẽ đảm bảo bộ PID luôn hoạt động trong vùng làm việc có sai số nhỏ, dẫn đến không cần sử dụng các phương pháp thay đổi nhiều bộ số PID để có được vùng hoạt động lớn như ở mục 6.

1. **Sử dụng các giải thuật thông minh để tìm ra các bộ số PID tối ưu:**

…

1. **PID kết hợp các giải thuật khác :**

PID thích nghi, PID-GA, PID-Fuzzy, …

**Cải tiến PID vị trí**

1. **Cải tiến các khâu của bộ PID truyền thống dạng công thức song song:**

**Đối với các biến điều khiển tích lũy (Hệ điều khiển vị trí góc quay của động cơ DC):**

1. **Cải tiến khâu I :**

Một hiện tượng thường gặp của khâu I là hiện tượng Wind – Up, hiện tượng này xảy ra khi mà sai số được tích trong phép tính này là quá lớn do nhiều nguyên nhân ví dụ như: Tích phân sai số thừa trong quá trình quá độ, tích phân sai số vô ích, …

Giải pháp: Chỉ để khâu tích phân hoạt động trong vùng hoạt động có ích của khâu này.

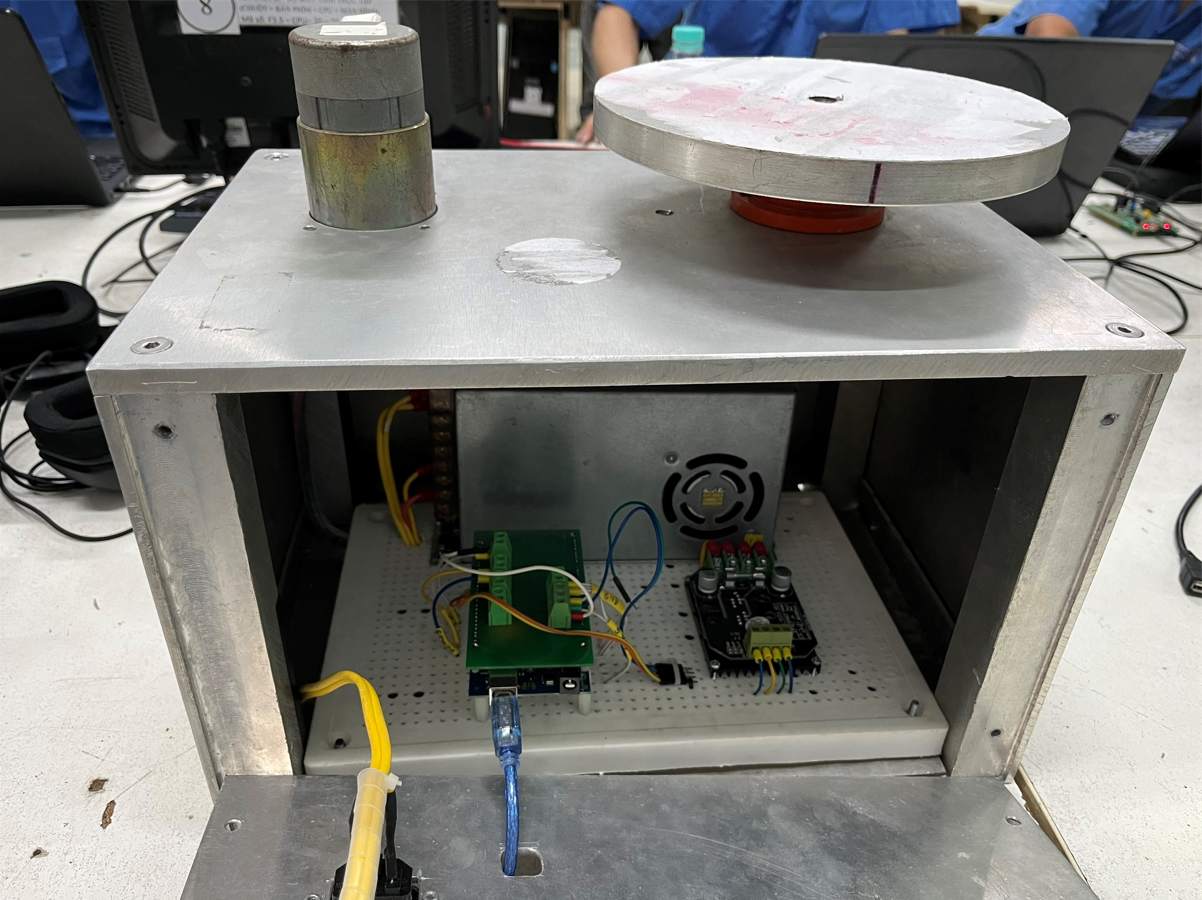
Giới hạn ngõ ra của khâu tích phân trong một dãy giá trị phù hợp.

Lúc này ta có thể điều chỉnh hệ số Ki lớn hơn giúp cho quá trình triệt tiêu sai số diễn ra nhanh hơn nhưng vẫn đảm bảo được tính chính xác. (Không xuất hiện vọt lố, đáp ứng sẽ không xuất hiện dao động thừa).

**Ví dụ** điều khiển vị trí một động cơ 24 VDC thông qua Matlab có sơ đồ phần cứng như sau:



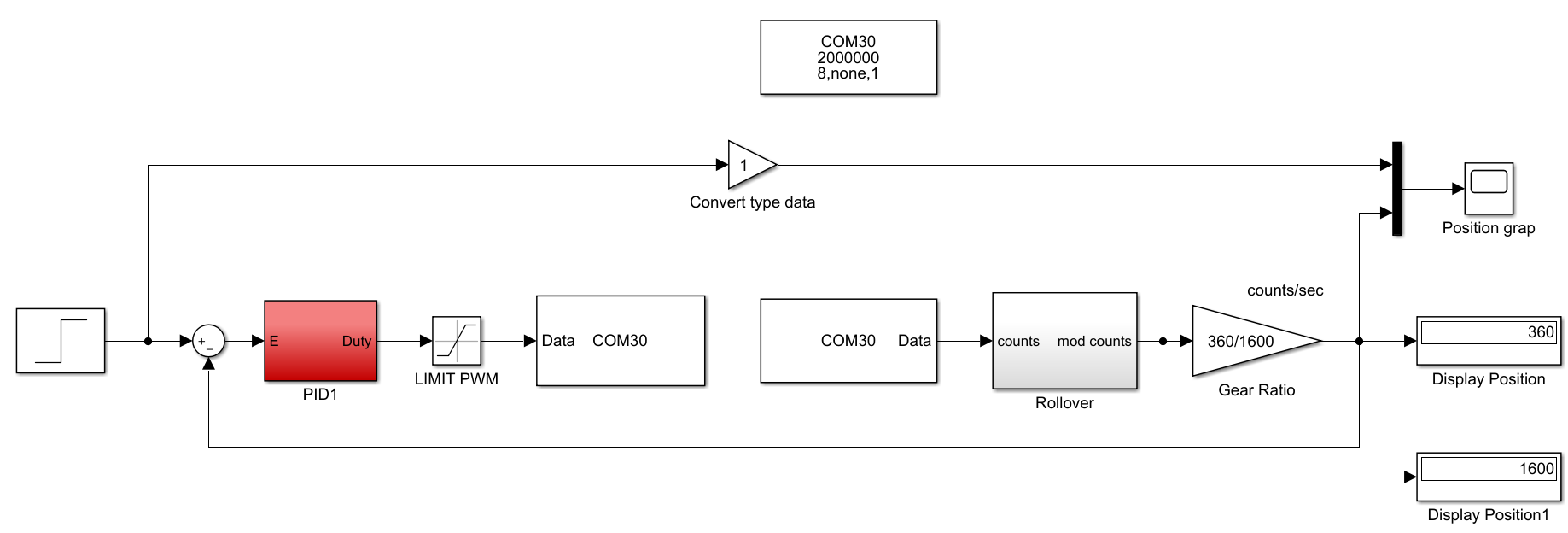
Sơ đồ kết nối phần cứng



Mô hình thực

Tần số PWM=4 [KHz], Ts=0.05 [s], Encoder 200 [xung/vòng], phương pháp đọc xung 2 kênh, 2 cạnh, giao tiếp UART tốc độ truyền nhận baud rate = 2.000.000, 8 bit/packet, none parity, tỷ số truyền puly là 1:2, xử lý dữ liệu nhận dạng interger 32 bit.

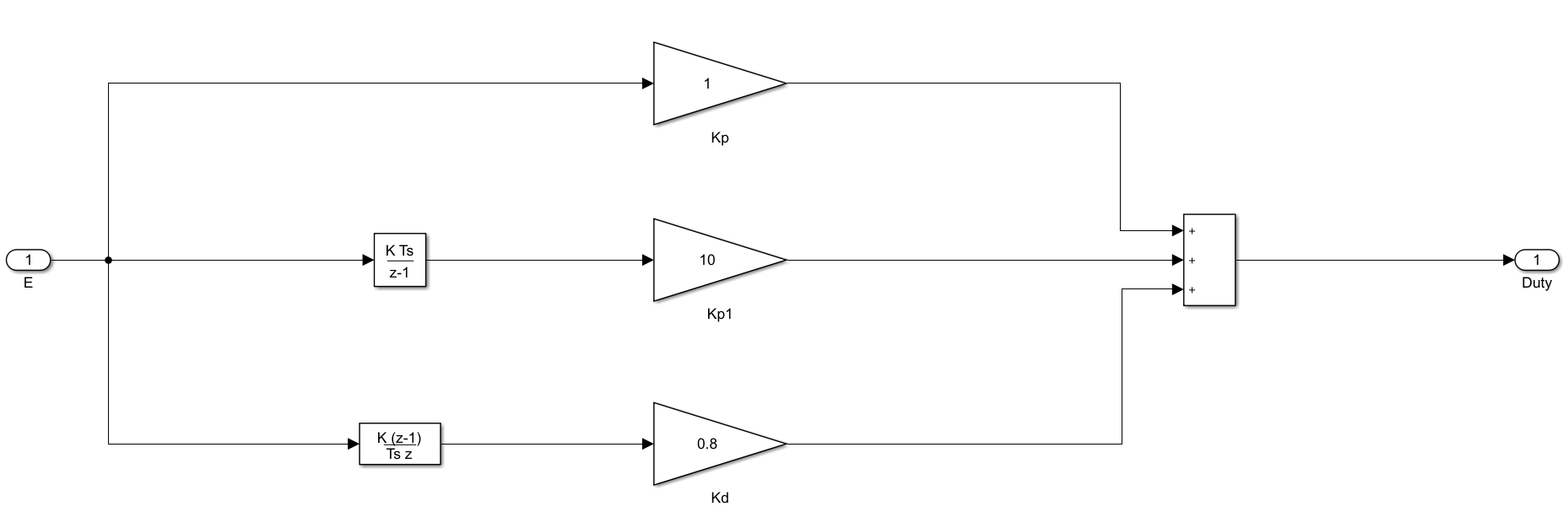
**Chương trình điều khiển bằng Matlab:**



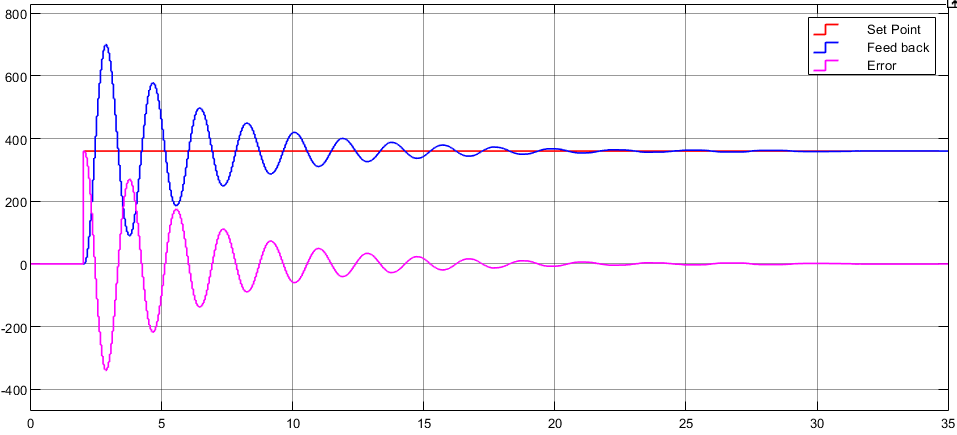
**1.PID truyền thống:**

Setpoint = 360 [độ], Steptime = 2 [s] sử dụng bộ PID song song theo công thức truyền thống và thu được đáp ứng như hình dưới:

Bộ PID:



Đáp ứng nấc Final Value = 360 [độ]

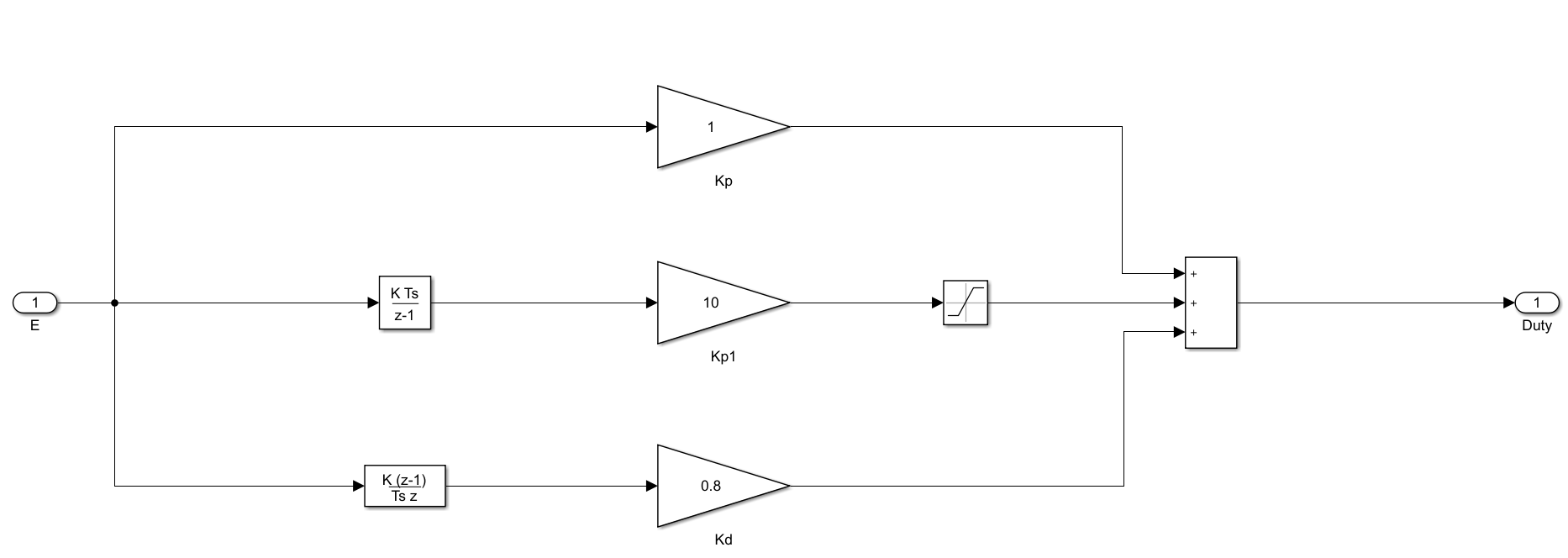


Nhận xét: Khâu I lúc này bị tích thừa sai số trong quá trình quá độ dẫn đến độ vọt lố lớn, tín hiệu bị dao động nhiều lần trước khi xác lập.

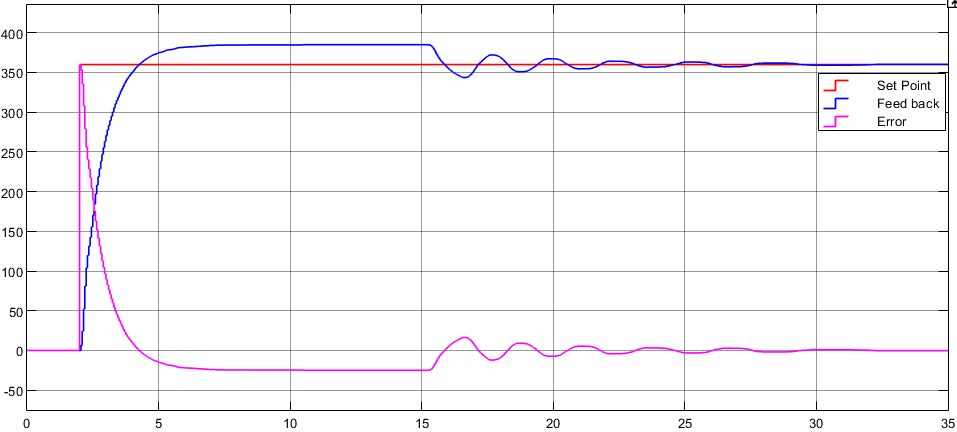
**2.PID giới hạn khâu I:**

Setpoint = 360 [độ], Steptime = 2 [s], ta thêm giới hạn khâu I từ [-40 40] để giảm một phần các đặc trưng không mong muồn của hiện tượng Wind – Up ta được đáp ứng như hình dưới.

Bộ PID có giới hạn khâu I:

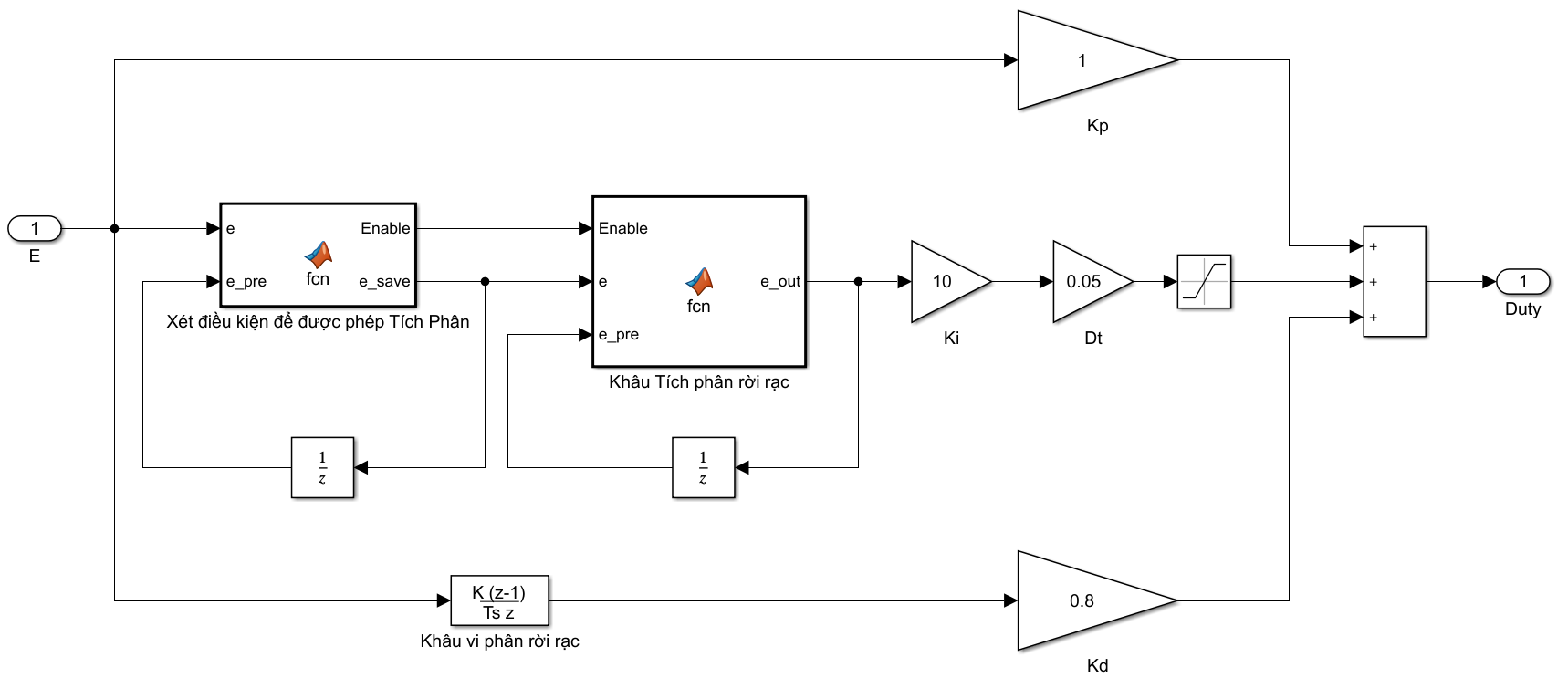


Đáp ứng nấc Final Value = 360 [độ]

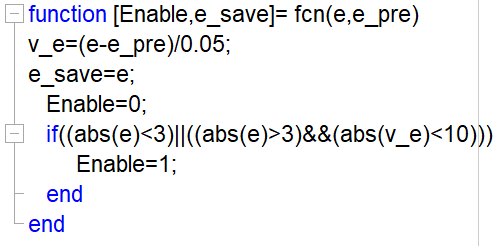


**3.PID có giới hạn khâu I và kiểm soát vùng hoạt động hiệu quả của khâu I:**

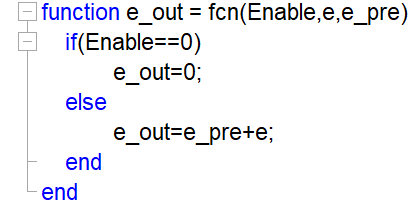
Setpoint = 360 [độ], Steptime = 2 [s], có khâu giới hạn khâu I từ [-40 40] và có kiểm soát vùng hoạt động hiệu quả của khâu I như hình dưới.



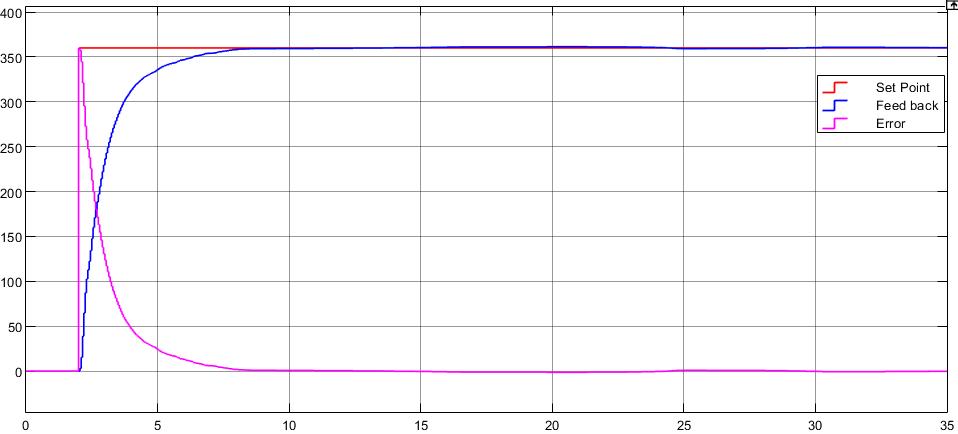
Khối xét điều kiện để được phép tích phân:

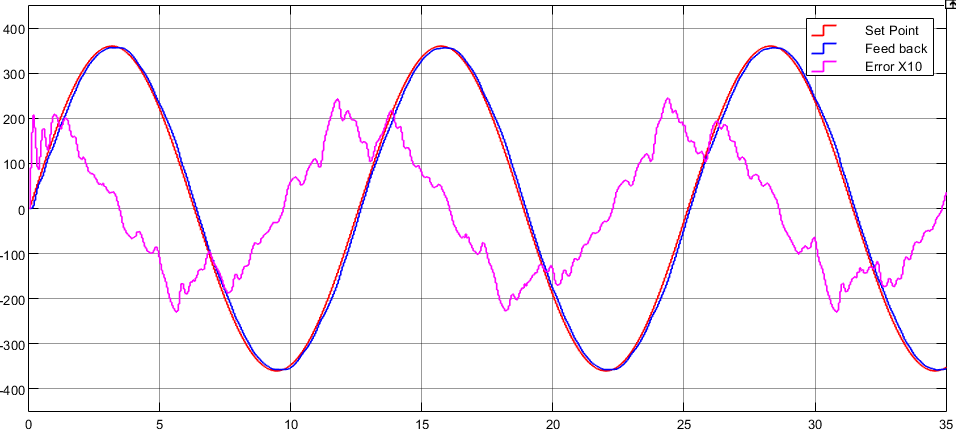


Khối tích phân:



Đáp ứng nấc Final Value = 360 [độ]

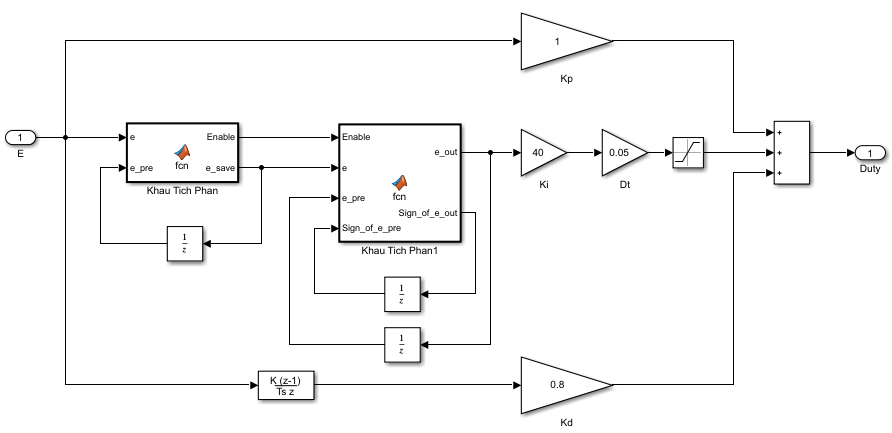




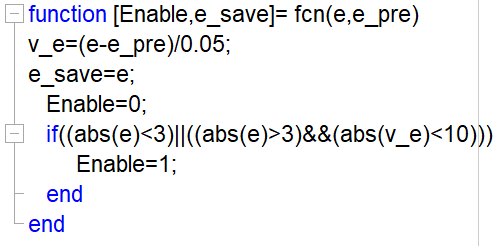
**Nhận xét:** Lúc này độ vọt lố giảm, Tổng của tích phân sai số dương và tích phân sai số âm trước khi xác lập giảm mạnh so với phương pháp cải tiến trước đó, thời gian quá độ giảm. Tuy nhiên nhược điểm của khâu I vẫn còn khi mà sai số đổi dấu thì toàn bộ giá trị tích phân trước thời điểm sai số đổi dấu sẽ tiếp tục làm tăng sai số nên ta phải bỏ phần sai số thừa này đi vào vào thời điểm này để có chất lượng đáp ứng tốt hơn, nếu làm như vậy ta có thể tăng Ki mà không bị tăng độ vọt lố.

**4. PID có giới hạn khâu I, kiểm soát vùng hoạt động hiệu quả của khâu I và loại bỏ giá trị tích phân thừa trong quá trình quá độ:**

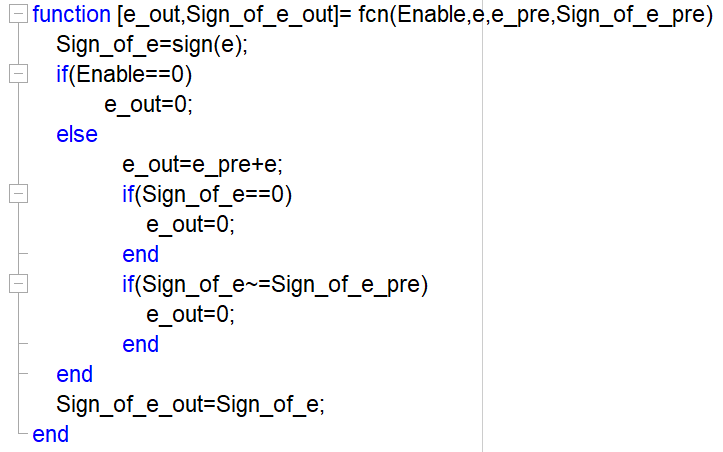
Bộ PID có giới hạn khâu I, kiểm soát vùng hoạt động hiệu quả của khâu I và loại bỏ giá trị tích phân thừa trong quá trình quá độ:



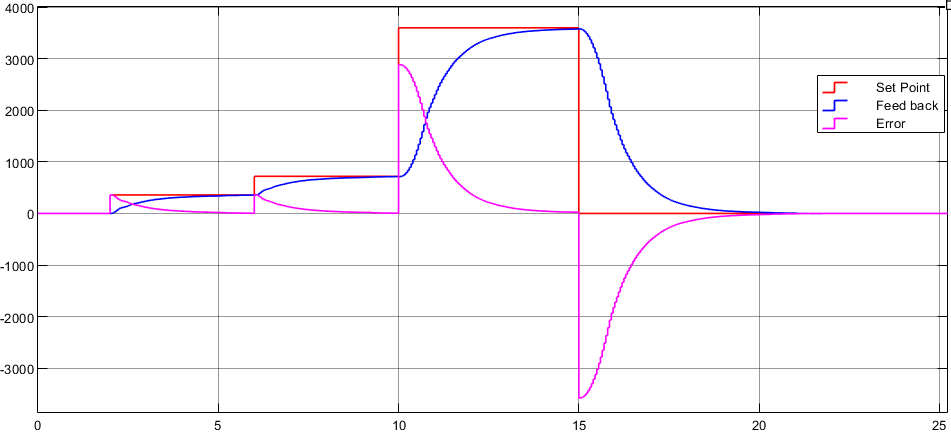
Khối xét điều kiện để được phép tích phân:



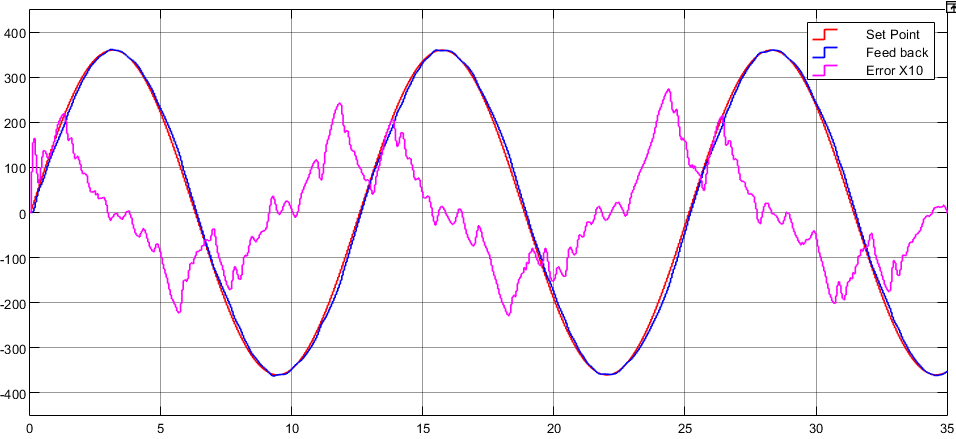
Khâu tích phân có kiểm soát điều kiện reset giá trị tích phân thừa:



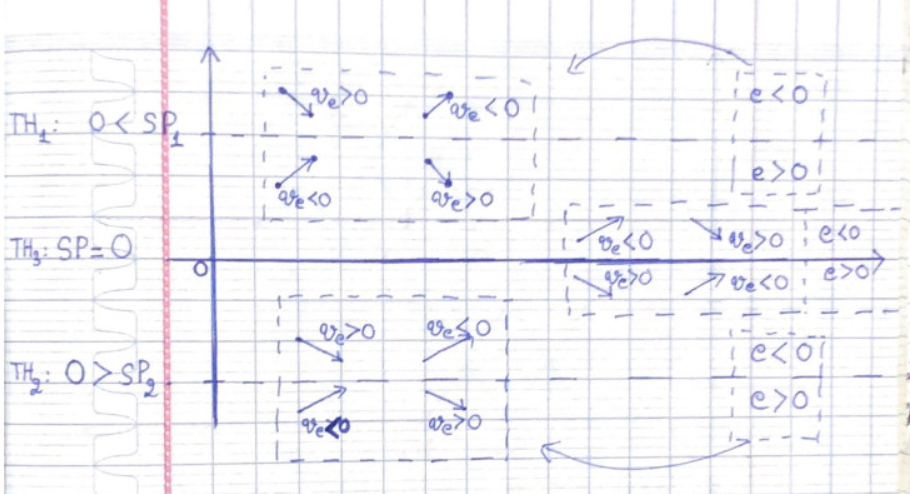
Đáp ứng nấc: Final Value = 360 [độ]



Đáp ứng tracking: Ngõ vào sóng Sin có biên độ là 360 và tần số là 0.5 [rad/s]



1. **Cải tiến khâu P và D:**

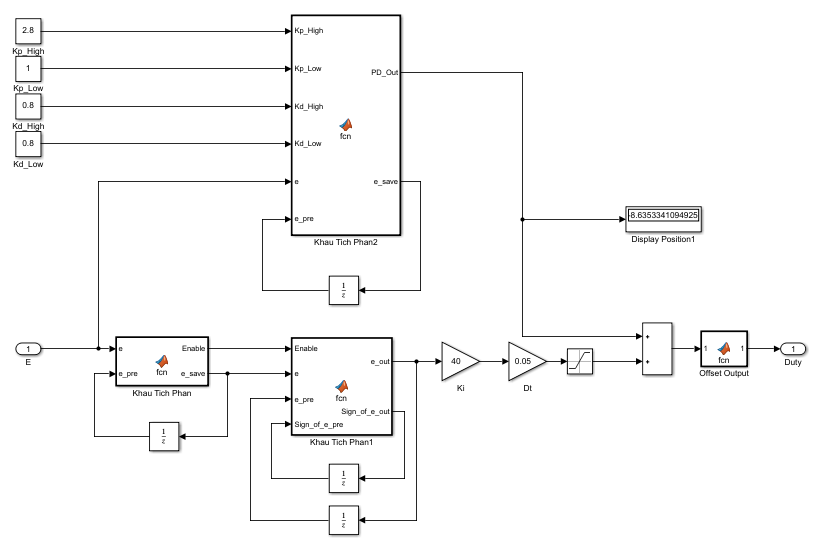


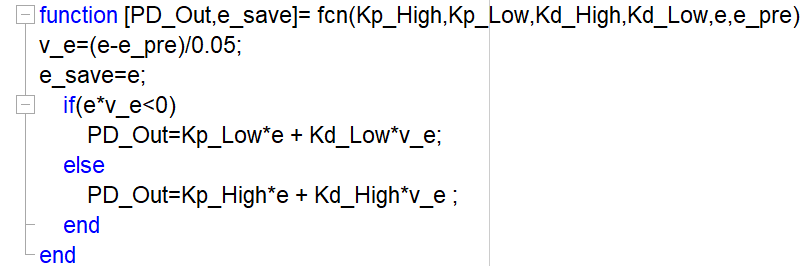
Vậy khi: thì sai số đang suy giảm.

thì sai số đang giữ nguyên giá trị.

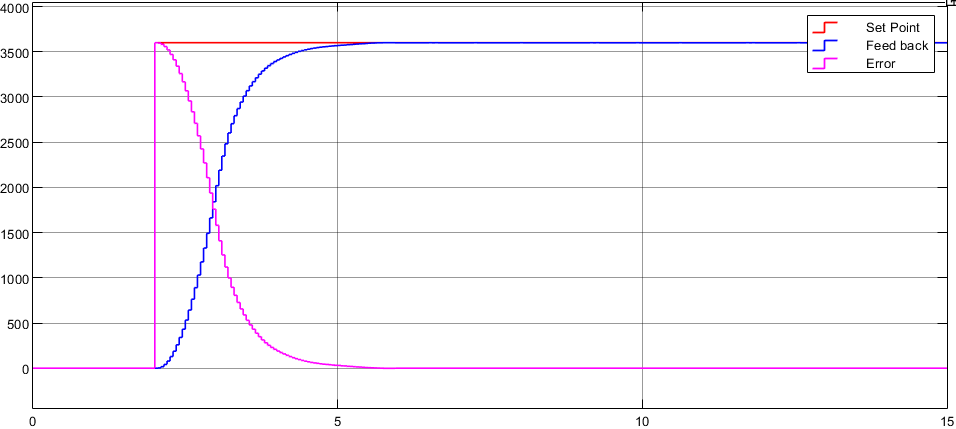
thì sai số đang tăng

Ta sẽ sử dụng các bộ thông số Kp và Kd thích hợp để cải thiện chất lượng đáp ứng tùy vào mỗi trường hợp trên

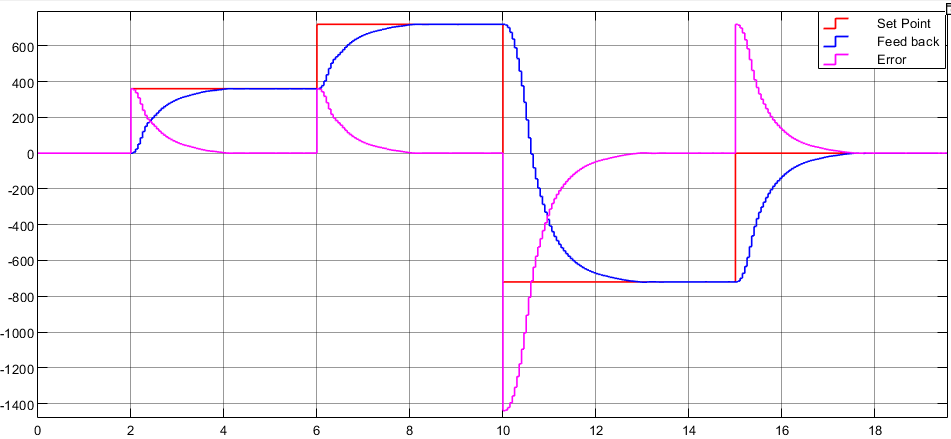




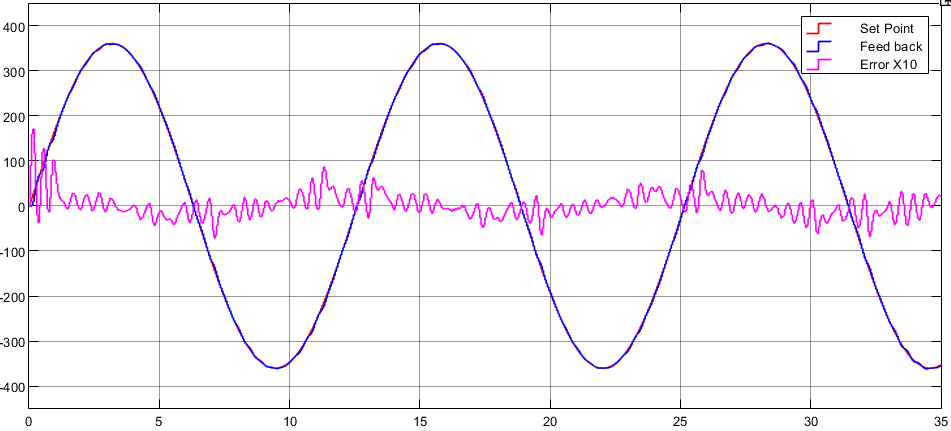
Đáp ứng nấc Final Value = 3600 [độ]



Đáp ứng nấc ở nhiều Setpoint và nhiễu xung do dùng tay đẩy bánh đà từ giây 25 đến giây 40



Đáp ứng tracking: Ngõ vào sóng Sin có biên độ là 360 và tần số là 0.5 [rad/s]



1. **So Sánh chất lượng:**

