**CHƯƠNG 4: Mô phỏng điều khiển động cơ trên Matlab / Simulink**

**4.1. Mục tiêu mô phỏng**

**4.1.1. Mô tả bài toán điều khiển BLDC**

Bài toán điều khiển động cơ BLDC đặt ra yêu cầu xây dựng một thuật toán điều khiển sao cho:

* Động cơ quay đúng theo yêu cầu đầu vào (về tốc độ và chiều quay).
* Quá trình chuyển mạch giữa các pha diễn ra chính xác dựa trên tín hiệu từ cảm biến vị trí.
* Hệ thống điều khiển có thể triển khai thực tế trên vi điều khiển nhúng.

Phương pháp điều khiển được lựa chọn trong mô hình này là 6-steps, một phương pháp phổ biến và đơn giản, thường dùng cho các hệ thống yêu cầu tốc độ trung bình đến cao mà không cần kiểm soát chính xác mô-men xoắn.

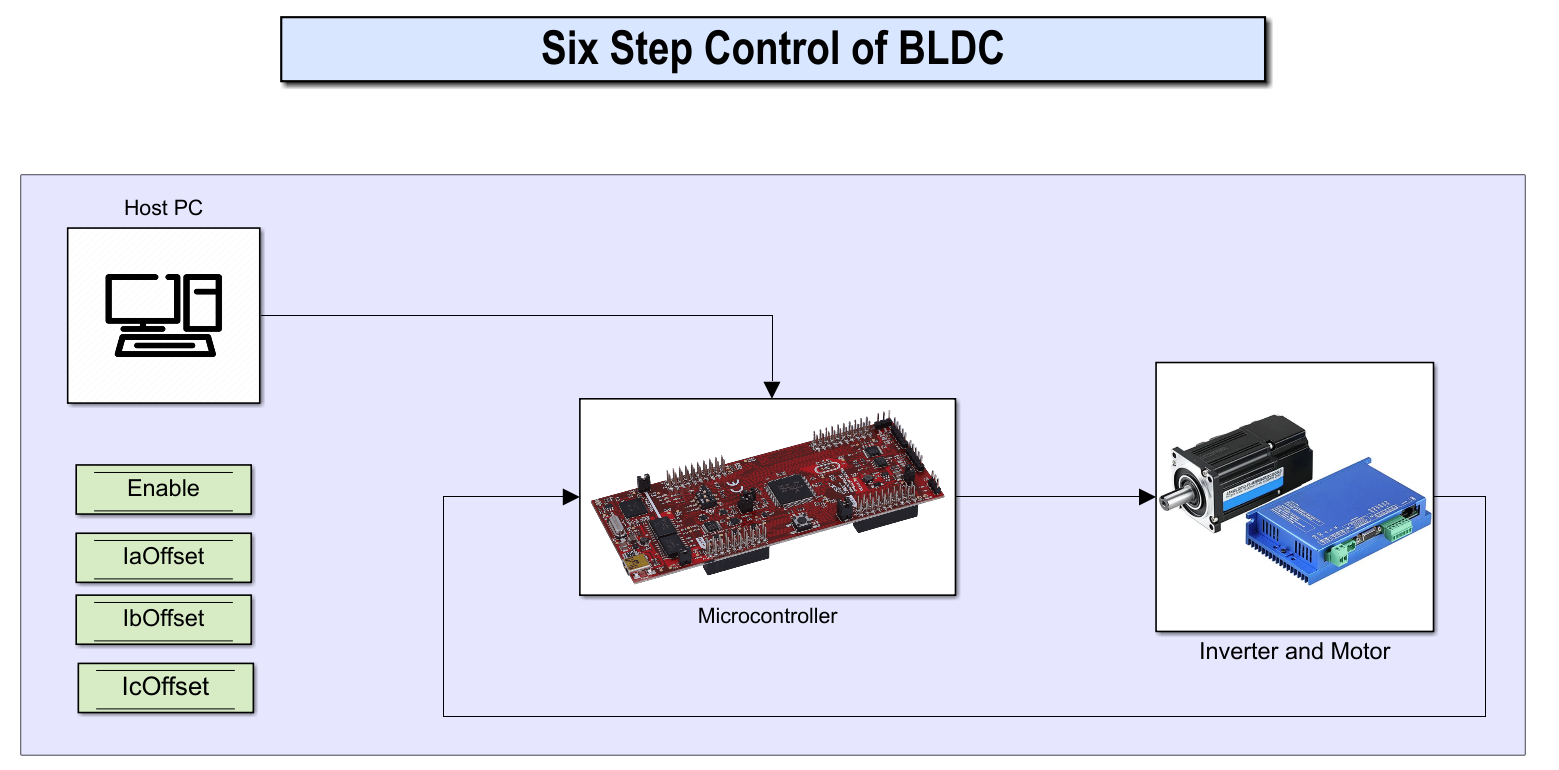
**4.1.2. Lý do sử dụng Matlab / Simulink và bộ công cụ hỗ trợ phần cứng**

MATLAB & Simulink là một môi trường mô phỏng mạnh mẽ, được sử dụng rộng rãi trong lĩnh vực điều khiển và nhúng, đặc biệt là các hệ thống điện – điện tử. Lý do sử dụng Simulink trong bài toán này bao gồm:

* Thiết kế trực quan: Simulink cung cấp giao diện kéo – thả (drag-and-drop) giúp dễ dàng xây dựng và hiểu mô hình hệ thống điều khiển.
* Mô phỏng động học: Có thể dễ dàng mô phỏng các tín hiệu dòng điện, điện áp, tốc độ và vị trí rotor theo thời gian.
* Tích hợp với phần cứng: Mô hình được thiết kế để tương thích với board LAUNCHXL-F28069M (dòng C2000 của Texas Instruments), hỗ trợ trực tiếp qua Embedded Coder và Motor Control Blockset, cho phép mô phỏng và triển khai code trên vi điều khiển thực.
* Dễ dàng kiểm thử và ghi nhận dữ liệu: Kết hợp với các công cụ như External Mode và Scope, mô hình cho phép theo dõi và ghi lại dữ liệu thời gian thực trong quá trình mô phỏng.

Việc sử dụng Simulink không chỉ giúp kiểm chứng trước giải thuật điều khiển mà còn rút ngắn thời gian triển khai thực tế, giảm thiểu sai sót trong quá trình viết code nhúng thủ công.

**4.2. Tổng quan mô hình mô phỏng**

****

Cấu trúc tổng quan của mô hình điều khiển động cơ BLDC 6-steps trên Simulink. Mô hình bao gồm các khối chức năng chính như:

* Host PC: giả lập tốc độ đặt từ host PC thông qua UART
* Microcontroller: bao gồm Speed Control - tạo dòng điện tham chiếu dựa trên sai số tốc độ và Current Control - điều chỉnh độ rộng xung PWM từ dòng tham chiếu và phản hồi dòng thực tế.
* Inverter and Motor – Plant Model: mô phỏng hoạt động của động cơ BLDC và mạch biến tần.

**4.3. Mô hình động cơ điện và biến tần – Plant Model**

**4.3.1. Mô hình động cơ điện**

Mô phỏng động cơ điện PMSM D60M-R6430 của hãng BeuDMKE với thông số:

|  |  |  |  |
| --- | --- | --- | --- |
| **Thông số** | **Kí hiệu** | **Giá trị** | **Đơn vị** |
| Công suất định mức | P\_rated | 200 | W |
| Số cặp cực | P | 5 | Cặp |
| Điện áp định mức | V\_rated | 48 | V |
| Dòng điện định mức | I\_rated | 5.1 | A |
| Dòng điện đỉnh | I\_peak | 10.2 | A |
| Tốc độ định mức | N\_rated | 3000 | rpm |
| Tốc độ tối đa | N\_max | 3200 | rpm |
| Momen xoắn định mức | T\_rated | 0.64 | N.m |
| Momen xoắn tối đa | T\_peak | 1.28 | N.m |
| Điện trở cuộn stator | Rs | 0.571 | Ω |
| Độ tự cảm trục d | Ld | 8.24e-4 | H |
| Độ tự cảm trục q | Lq | 8.24e-4 | H |
| Quán tính | J | 3e-5 | kg.m2 |
| Hệ số ma sát | B | 2.6368752178e-06 | kg.m2/s |
| Hệ số sức điện động | Ke | 7.5 | V/krpm |
| Hệ số momen xoắn | Kt | 0.12 | N.m/A |
| Kiểu quấn dây |  | Wye wounded |  |
| Encoder QEPSlits |  | 2500 | PPR |
| Encoder offset |  | 0.1995 |  |

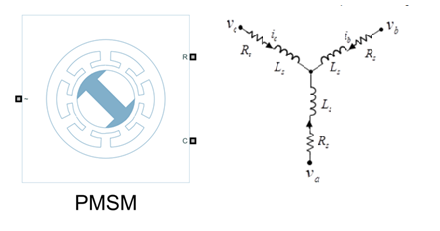
Ngoài ra cần tính toán Liên Kết Dòng Từ Của Nam Châm Vĩnh Cửu (PM Flux Linkage) – vốn chưa được cung cấp bởi nhà sản xuất động cơ điện:

 [1]

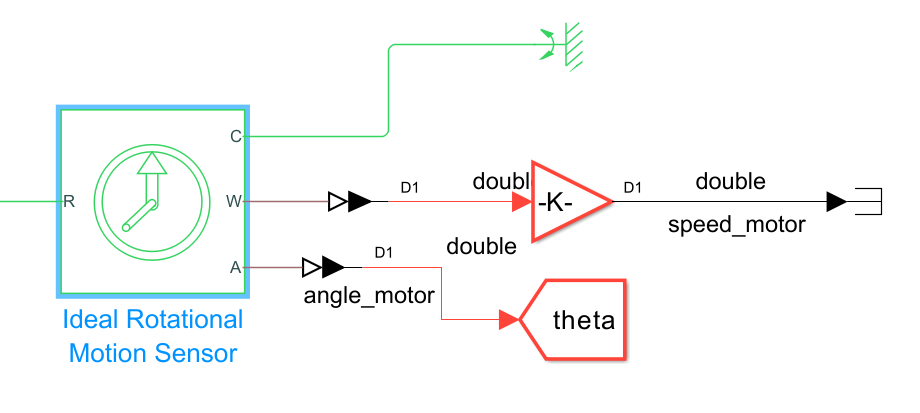
Trong đó:

* *Ke* : hằng số suất điện động ngược (V/krpm)
* *p*: số cặp cực của động cơ điện

Sau khi thu thập đầy đủ thông số động cơ, các giá trị này được lưu vào biến trong script MATLAB. Sử dụng khối PMSM trong Simscape Electrical nhận dòng ba pha từ inverter và mô phỏng phản ứng điện. Khối này hỗ trợ giám sát các đại lượng như tốc độ, mô-men, vị trí và dòng cuộn dây, và đặc biệt phù hợp cho việc kiểm thử các thuật toán điều khiển như FOC, DTC hoặc 6-step.



Sử dụng khối Ideal Rotational Motion Sensor kết nối đến cổng “R” của PMSM, lấy thông tin về vị trí của rotor, tín hiệu đầu ra là góc quay của rotor (rad)

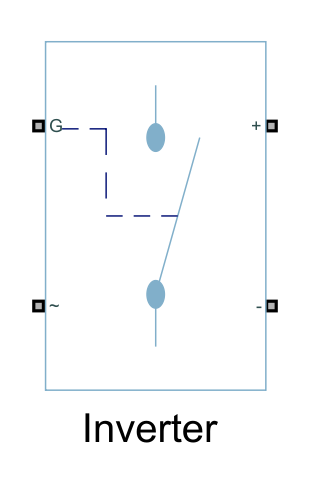


**4.3.2. Mô hình biến tần (Inverter)**

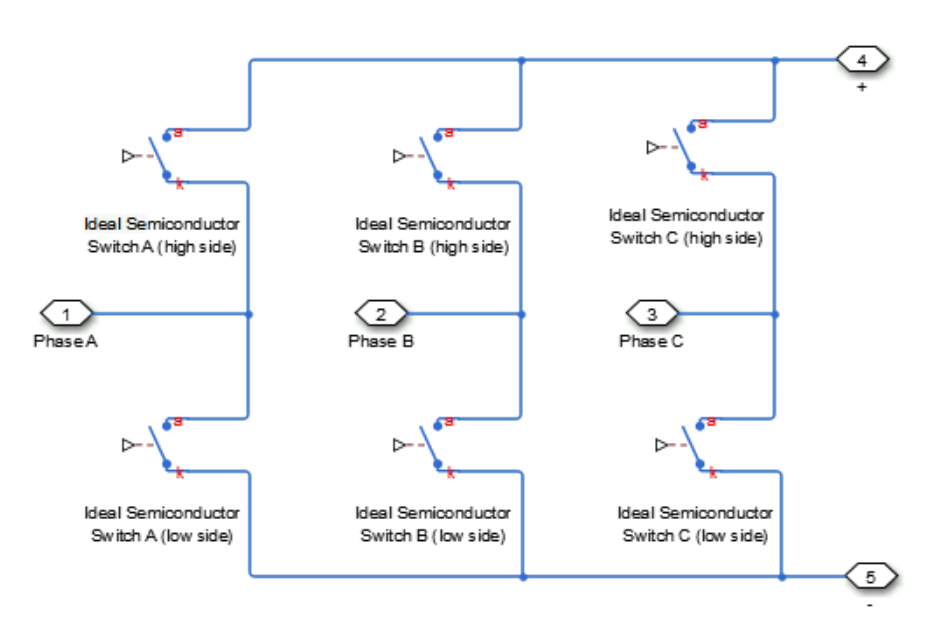
Biến tần được sử dụng là SBLMB500, kết hợp với cảm biến dòng điện WCS1800, có thông số như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Thông số** | **Kí hiệu** | **Giá trị** | **Đơn vị** |
| Điện áp làm việc | V\_dc | 48 | V |
| Ngưỡng bảo vệ quá dòng | I\_trip | 20 | A |
| Điện trở dẫn của MOSFET | Rds\_on | 10.3e-3 | Ω |
| Điện trở Shunt đo dòng | R\_shunt | - | Ω |
| Offset cảm biến dòng pha A | CtSensAOffset | 3309 |  |
| Offset cảm biến dòng pha B | CtSensBOffset | 3003 |  |
| Offset cảm biến dòng pha C | CtSensCOffset | 3318 |  |
| Hệ số khuếch đại tín hiệu ADC | ADCgain | 1 |  |
| Logic điều khiển | EnableLogic | 1 |  |
| Đảo cực khuếch đại tín hiệu | InvertingAmp | - |  |
| Điện áp tham chiếu cảm biến dòng | ISenseVref | 5 | V |
| Độ nhạy cảm biến dòng | ISenseVoltPerAmp | 0.06 | V/A |
| Ngưỡng đo dòng tối đa | IsenseMax | 35 | A |

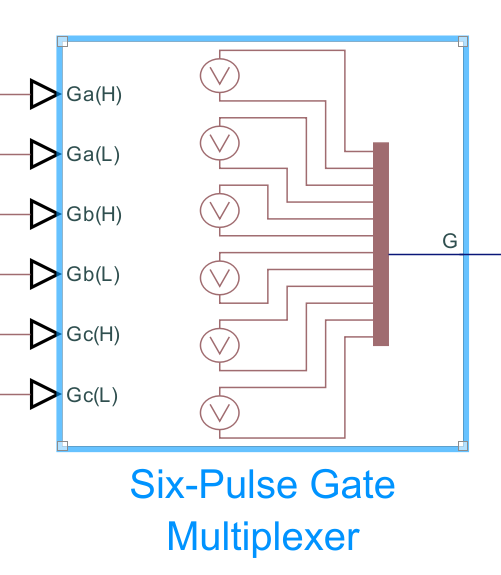
Sau khi có đầy đủ thông số của bộ biến tần, ta sử dụng khối Inverter (Three-phase) trong thư viện Simscape để mô phỏng lại phần cứng được sử dụng.



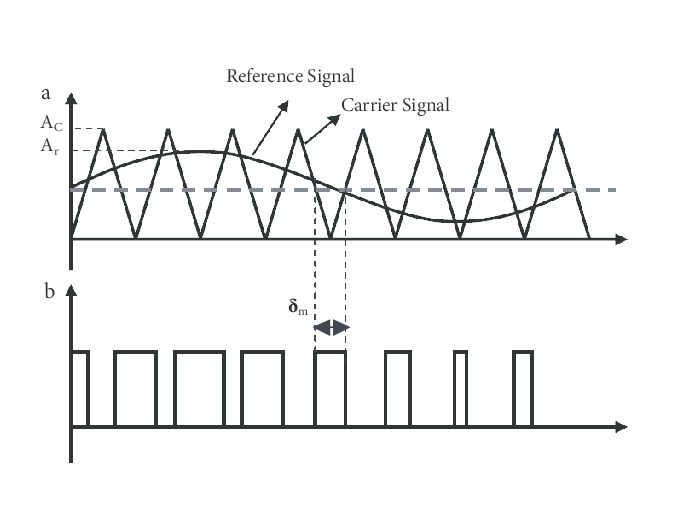
Hình sau đây mô tả mạch tương đương cho khối Inverter với đầy đủ các công tắc bán dẫn (MOSFET) được điều khiển độc lập, bao gồm ba nhánh pha (A, B, C), mỗi nhánh gồm hai MOSFET mắc nối tiếp: một phía trên (high side) và một phía dưới (low side), tạo thành cấu trúc cầu ba pha (three-arm bridge). Giữa mỗi cặp MOSFET là điểm trung tính của pha, được nối đến mỗi pha của động cơ điện. Hai cực trên – dưới của bộ Inverter lần lượt nối với nguồn DC. [2]



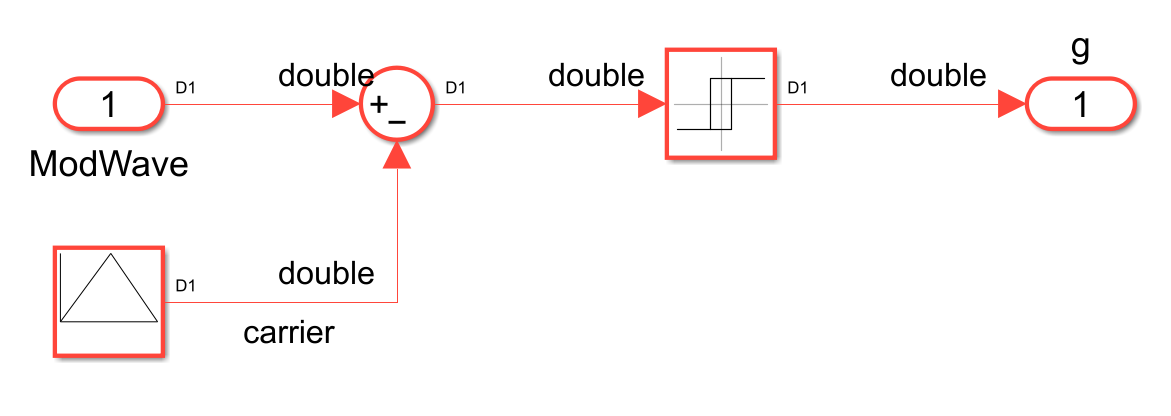
Vì có 6 MOSFET cần được điều khiển nhưng chỉ có một đầu vào của khối Inverter được đánh dấu “G”, vì vậy để giải quyết điều này, một khối dựa trên Simscape có tên là "SixPulse Gate Multiplexer" được sử dụng. Khối này gộp 6 tín hiệu điều khiển MOSFET từ bên ngoài thành một vector đơn, sau đó được kết nối đến cổng “G” của Inverter.



Để tạo được tín hiệu PWM điều khiển MOSFET, vi điều khiển (MCU) dựa trên cơ chế so sánh giữa một tín hiệu điều khiển (Duty Cycle) và một sóng mang (Carrier Wave), thường là sóng tam giác hoặc răng cưa được sinh ra nội bộ từ bộ định thời (Timer). Sóng mang này có tần số cố định và được tạo ra bằng cách đếm lên/xuống liên tục trong khoảng giá trị xác định. Giá trị Duty Cycle thường được tính toán từ bộ điều khiển (như PI hoặc PID), phản ánh mức năng lượng cần cấp cho động cơ và được lưu vào thanh ghi so sánh của vi điều khiển. Tại mỗi chu kỳ đếm, nếu giá trị duty lớn hơn giá trị đếm hiện tại của sóng mang, đầu ra PWM sẽ ở mức cao (ON); ngược lại là mức thấp (OFF). Kết quả là một tín hiệu vuông có độ rộng xung thay đổi theo nhu cầu điều khiển. Phương pháp này đảm bảo tín hiệu PWM được tạo ổn định và có thể điều khiển chính xác dòng điện đi qua cung cấp cho động cơ.

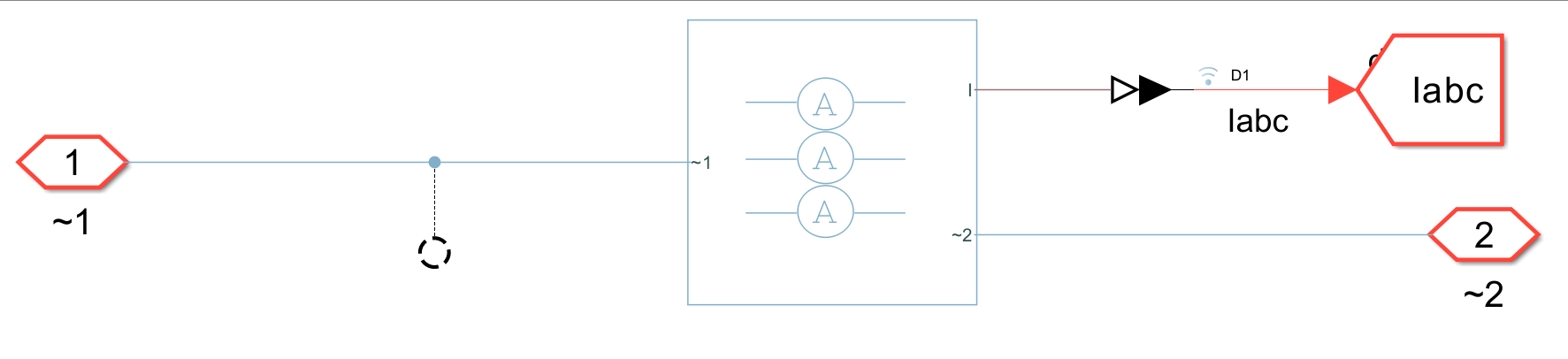


*Cách tạo PWM từ duty cycle và carrier wave*



*Khối tạo PWM trong mô hình*

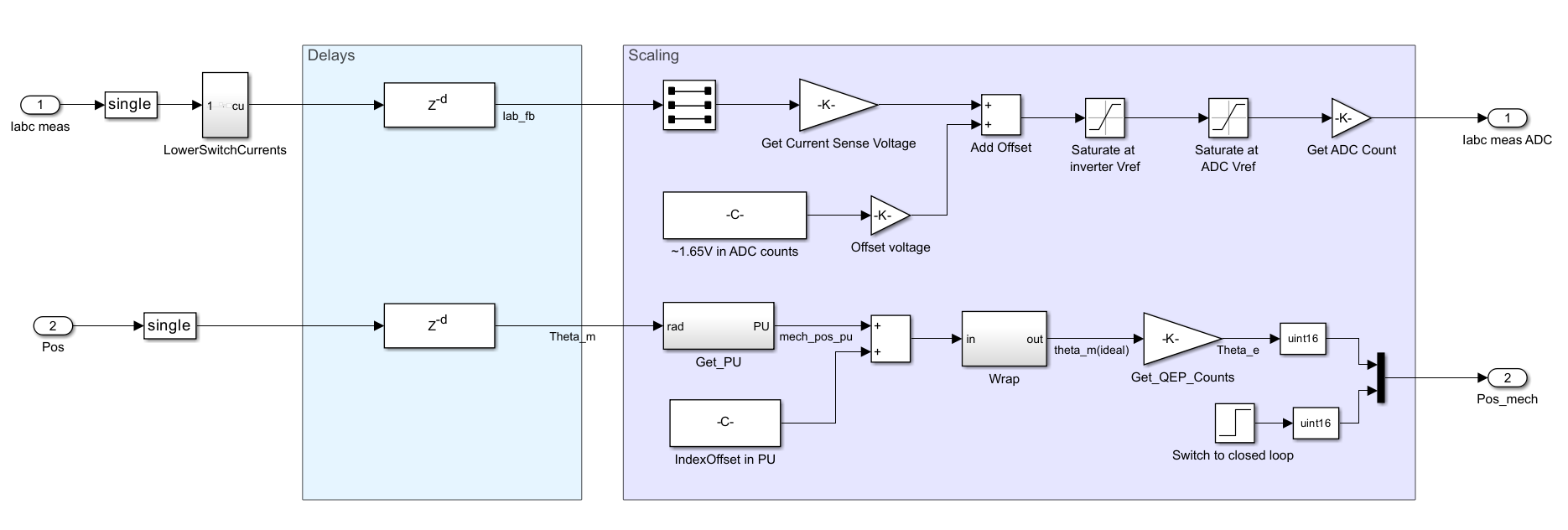
Sau khi nhận được tín hiệu điều khiển MOSFET, bộ inverter cung cấp dòng điện đến các pha của động cơ điện. Dòng điện này phải được đo và phản hồi về cho vi điều khiển để phục vụ cho việc kiểm soát động cơ.



*Sử dụng khối “Current Sensor (Three-Phase)” để đo dòng điện qua động cơ*

**4.3.3. Khối giả lập cảm biến**

Trong hệ thống điều khiển động cơ điện, đặc biệt trên phần cứng thực tế, một vấn đề quan trọng cần được xử lý là cách thức thu nhận và diễn giải tín hiệu phản hồi từ động cơ. Các cảm biến vật lý như cảm biến dòng (current sensor) hay cảm biến vị trí (encoder) không cung cấp trực tiếp các giá trị như dòng điện (A) hoặc góc quay rotor (rad). Thay vào đó, các cảm biến này chỉ xuất ra tín hiệu ở dạng analog (tín hiệu điện áp) hoặc digital (dạng xung hoặc mã hóa số) mang tính gián tiếp và cần phải được xử lý thêm. Vi điều khiển cần kết hợp thêm với các khối xử lý như bộ khuếch đại, bù offset, lấy mẫu ADC và sau đó chuyển đổi sang giá trị dòng điện tương ứng (đơn vị A). Tương tự, tín hiệu từ encoder chỉ là các xung vuông pha A/B hoặc giá trị đếm (count), nên cần phải qua các khối xử lý như thuật toán đếm xung, bộ chia độ phân giải, hoặc giải mã góc quay để suy ra vị trí rotor theo đơn vị radian (rad) hoặc độ (°). Vì lý do đó, trong mô hình điều khiển luôn cần có các khối trung gian để chuyển đổi tín hiệu cảm biến sang dạng thông tin chuẩn hóa mà hệ thống có thể hiểu và khai thác. Những khối này có thể được gọi là các khối chuyển đổi tín hiệu hoặc xử lý tín hiệu cảm biến (sensor signal conditioning), giữ vai trò quan trọng trong việc kết nối mô hình vật lý (analog/digital) với bộ xử lý số (digital controller) nằm trong mô hình mô phỏng.



*Sơ đồ khối giả lập cảm biến*

Nhánh trên của sơ đồ minh họa quy trình chuyển đổi tín hiệu dòng điện thành giá trị ADC mà vi điều khiển có thể đọc được. Đầu tiên, khối LowerSwitchCurrents được sử dụng để lọc bỏ thành phần dương, giữ lại phần dòng âm – tương ứng với dòng chảy qua các MOSFET lowside của Inverter. Tiếp theo là khối tạo độ trễ một chu kỳ (z⁻¹), được đưa vào nhằm mô phỏng hai loại độ trễ thực tế: độ trễ tính toán trong vi điều khiển khi cập nhật PWM, và độ trễ vật lý do thời gian chuyển mạch MOSFET – thường tương đương nửa chu kỳ PWM [3]. Trong mô hình này, độ trễ được chọn là 1 chu kỳ. Sau đó, dòng điện được nhân với hệ số tỷ lệ điện áp trên dòng (Volt/Ampere), đại diện cho độ nhạy của cảm biến dòng – trong trường hợp này là 0.07 V/A, được cấu hình trong khối *Inverter configuration*. Giá trị điện áp thu được sẽ được cộng thêm điện áp offset (tại ~ 1.65V), nhằm mô phỏng điểm không lý tưởng của mạch đo. Để bảo vệ hệ thống và giữ giá trị điện áp trong dải hợp lệ, kết quả sau đó được giới hạn (saturate) tại hai ngưỡng: điện áp tham chiếu inverter (5V) và điện áp tham chiếu ADC (3.3V). Dòng điện tối đa có thể đo được được coi là dòng điện cơ sở (base). Giá trị ADC có thể được tính toán từ dòng điện cơ sở và độ phân giải bộ ADC (12bit), cùng với ADC offset, bằng cách sử dụng phương trình:

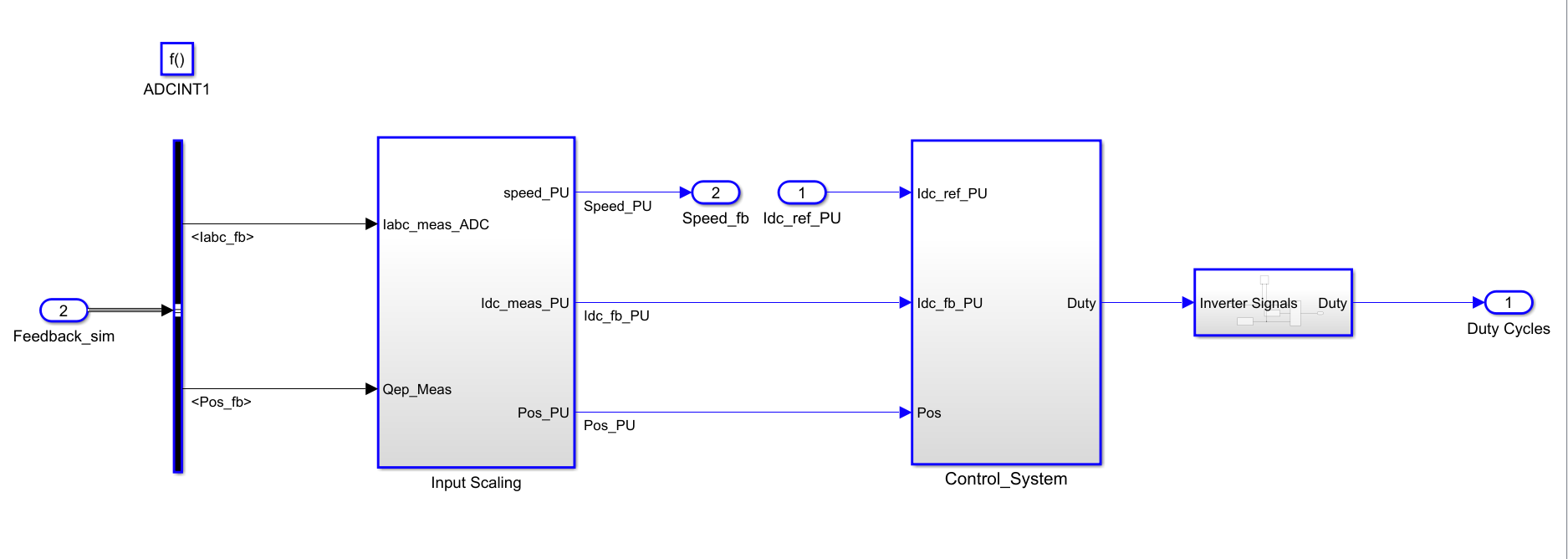
[4]

Nhánh dưới của sơ đồ thể hiện quá trình chuyển đổi tín hiệu vị trí rotor từ encoder sang dạng giá trị số (Pos\_mech) mà vi điều khiển có thể sử dụng trong thuật toán điều khiển. Tín hiệu đầu vào ban đầu (Pos) thường là một giá trị góc quay liên tục, được mô phỏng hoặc thu từ mô hình động học của động cơ. Trước tiên, tín hiệu này được đưa qua khối độ trễ một chu kỳ (z⁻¹) nhằm đồng bộ thời gian và mô phỏng độ trễ thực tế trong thu nhận và xử lý tín hiệu encoder. Tiếp theo, tín hiệu góc được chuyển đổi sang đơn vị per-unit (PU) thông qua khối Get\_PU, tức là được chuẩn hóa trong khoảng từ 0 đến 1 theo một vòng quay đầy đủ. Nếu có sai số lắp đặt hoặc lệch zero từ encoder, một giá trị offset góc cũng được bù trừ thông qua khối IndexOffset in PU. Sau bước chuẩn hóa, tín hiệu được đưa vào khối Wrap, có nhiệm vụ "quấn vòng" giá trị góc để đảm bảo nó luôn nằm trong khoảng [0, 1] hoặc [0, 2π], tránh bị tràn khi quay liên tục. Kết quả sau đó được nhân với tổng số xung của encoder (QEPSlits) trong khối Get\_QEP\_Counts, chuyển từ giá trị PU sang số đếm encoder thực tế (Counts). Cuối cùng, tín hiệu được định dạng sang kiểu unsigned 16-bit (uint16) để phù hợp với dạng dữ liệu đầu vào của vi điều khiển, đồng thời được kết nối với điều kiện chuyển sang điều khiển vòng kín thông qua khối Switch to closed loop. Đầu ra Pos\_mech là một giá trị góc đã được xử lý, số hóa và chuẩn hóa, đại diện cho vị trí tức thời của rotor.

**4.4. Mô hình khối Microcontroller**

**4.4.1. Khối điều khiển dòng điện – Current control**

Khối Current Control đóng vai trò trung tâm trong vòng điều khiển bên trong (inner loop) của hệ thống. Nhiệm vụ chính của nó là điều chỉnh độ rộng xung PWM (Duty Cycles) để đảm bảo dòng điện cung cấp cho động cơ bám sát theo giá trị tham chiếu (Idc\_ref\_PU) được tính từ bộ điều khiển tốc độ (Speed Control). Khối này nhận vào tín hiệu phản hồi từ mô hình động cơ (Plant Model) dưới dạng feedback\_sim, sau đó thực hiện xử lý tín hiệu cảm biến, chuẩn hóa dòng điện và vị trí rotor và cuối cùng sử dụng những thông tin đó để tạo xung điều khiển nghịch lưu (inverter).



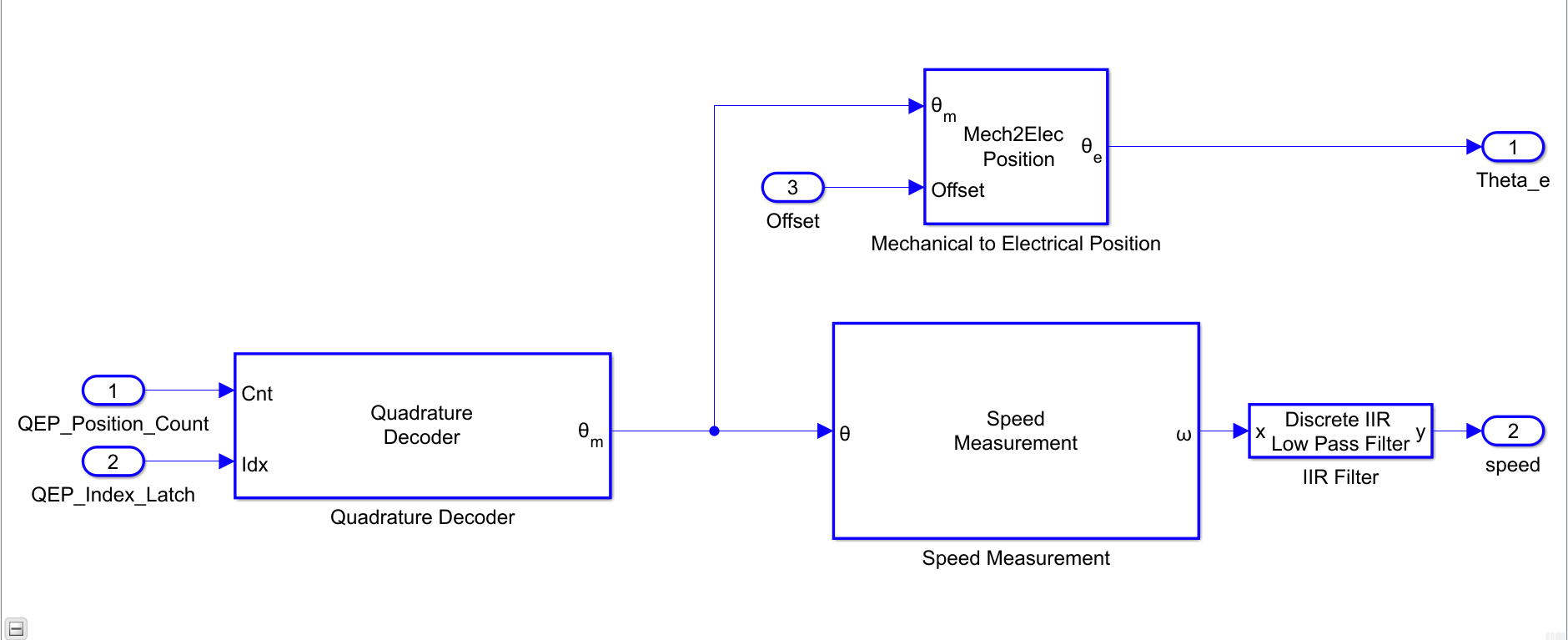
**4.4.1.1. Khối chuyển đổi tín hiệu – Input Scaling**

Trong hệ thống điều khiển động cơ, các tín hiệu phản hồi từ cảm biến (dòng điện, tốc độ, vị trí...) thường ở dạng tín hiệu điện áp hoặc số đếm và không trực tiếp mang đơn vị vật lý chuẩn (A, rad/s, rad). Những tín hiệu này phải được chuyển đổi thành đơn vị chuẩn hóa, dễ xử lý và độc lập với dải đo của phần cứng. Trong đó các đại lượng đo sẽ được chuyển về đơn vị PU. PU (per-unit) là một hệ đơn vị chuẩn hóa, trong đó tất cả các giá trị được biểu diễn dưới dạng tỷ lệ so với một giá trị cơ sở (base value) theo công thức:



Với các thuận lợi trong tính toán như: đơn giản hóa tính toán điều khiển, giúp thuật toán làm việc hiệu quả hơn vì mọi đại lượng đều nằm trong khoảng giới hạn chuẩn (thường [0–1] hoặc [−1, 1]); Tăng tính tương thích phần cứng, không cần thay đổi thuật toán nếu đổi sang động cơ khác – chỉ cần cập nhật thông số cơ sở. Việc sử dụng Input Scaling sang PU giúp chuyển đổi các tín hiệu vật lý đầu vào từ cảm biến thành dữ liệu chuẩn hóa để hệ điều khiển xử lý chính xác, hiệu quả và tương thích tốt với nhiều loại động cơ và phần cứng khác nhau.

Với tín hiệu encoder, việc chuyển đổi được thực hiện theo sơ đồ sau:



Trong hệ thống điều khiển động cơ BLDC sử dụng encoder, quá trình xử lý tín hiệu phản hồi từ encoder được thực hiện qua ba bước chính: giải mã xung vị trí, chuyển đổi góc cơ học sang góc điện, và tính toán tốc độ quay. Đầu tiên, khối Quadrature Decoder tiếp nhận tín hiệu từ encoder dưới dạng số đếm (*QEP\_Position\_Count*) và tín hiệu đánh dấu chỉ vị trí gốc (*QEP\_Index\_Latch*). Từ đây, khối tính toán ra góc quay cơ học *θm* theo công thức:



Trong đó PPR là số rãnh của encoder (2500). Kết quả là một góc quay liên tục trong khoảng [0, 2π], đại diện cho một vòng quay cơ học của rotor.

Vì trong điều khiển điện, các tín hiệu dòng và điện áp phải được đồng bộ với chu kỳ điện từ, không phải với chu kỳ cơ học, khối Mech2ElecPosition thực hiện chuyển đổi góc cơ học ​thành góc điện. Góc điện là một khái niệm quan trọng trong điều khiển động cơ đồng bộ (như BLDC hoặc PMSM), đại diện cho vị trí tương đối của từ trường rotor so với từ trường stator trong hệ quy chiếu điện, giúp đồng bộ hóa thời điểm đóng/mở các van bán dẫn với vị trí thực tế của rotor. Để thực hiện chuyển đổi góc cơ học thành góc điện, sử dụng công thức:



Trong đó:

* P là số cực của động cơ
* Offset là hệ số hiệu chỉnh để bù sai lệch lắp đặt giữa trục rotor và encoder.

Song song với đó, tín hiệu cũng được đưa vào khối Speed Measurement, nơi tính đạo hàm theo thời gian để thu được tốc độ góc ω.Để tính tốc độ quay, ta lấy đạo hàm của θₘ theo thời gian:



Chuyển đổi tốc độ góc cơ khí thành tốc độ góc điện theo công thức



Để ổn định và loại nhiễu cao tần trong quá trình đo, tốc độ này được làm mượt bởi khối IIR Low Pass Filter, cho ra kết quả cuối cùng là *Speed\_PU.*

Tiếp theo, khối chuyển đổi ADC sang PU có nhiệm vụ xử lý tín hiệu dòng điện ba pha đo được từ cảm biến dòng (qua ADC) để tạo ra dòng điện phản hồi chuẩn hóa, dùng trong vòng điều khiển dòng. Tín hiệu đầu vào Iabc\_meas\_ADC ban đầu là số nguyên 16-bit được chuyển sang kiểu int32 nhằm tăng độ chính xác trong các phép tính toán số. Sau đó, giá trị ADC này được trừ đi offset, tương ứng với mỗi pha (IaOffset, IbOffset, IcOffset), để loại bỏ sai số cố định trong phần cứng đo dòng. Công thức tính giá trị ADC sau hiệu chỉnh là:

IADC\_adj = IADC - ADCOffset

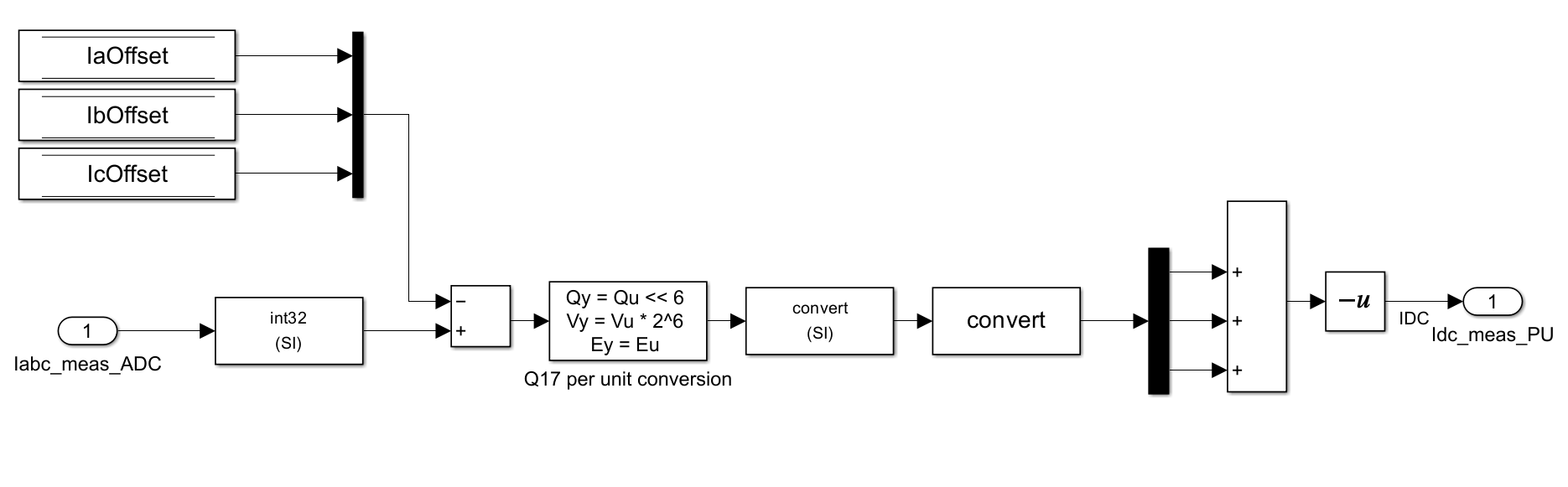
Tiếp theo, khối “Q17 per unit conversion” sẽ chuẩn hóa giá trị ADC đã hiệu chỉnh sang dạng Q-format, cụ thể là Q17, bằng cách dịch trái 6 bit. Phép nhân này giúp tăng độ phân giải trong biểu diễn số nguyên cố định, phục vụ cho các thuật toán điều khiển trong vi điều khiển số (digital controller):



Giá trị này sau đó được chuyển đổi sang kiểu số thực (floating-point) để tiếp tục sử dụng trong các phép tính điều khiển ở dạng chuẩn hóa hoặc trong các khối mô phỏng không hỗ trợ Q-format trực tiếp. Sau khi chuyển đổi, ba dòng điện của ba pha được cộng lại để tính dòng DC trung bình, giả định rằng động cơ hoạt động với các dòng pha cân bằng. Công thức cộng ba dòng pha là:

Idc = Ia + Ib + Ic

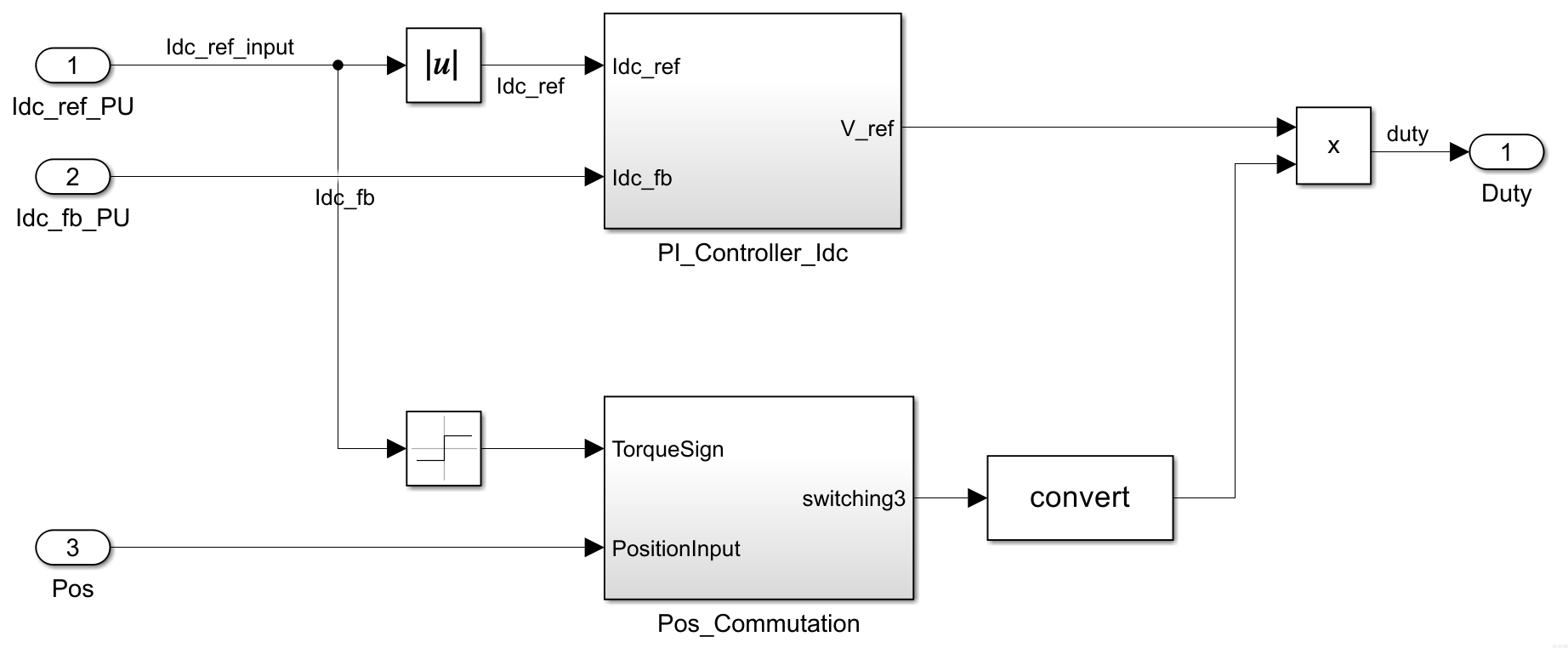
Kết quả cuối cùng là đầu ra Idc\_meas\_PU, đại diện cho dòng điện tổng đã được chuẩn hóa theo đơn vị PU. Việc chuẩn hóa dòng theo PU giúp thuật toán điều khiển hoạt động ổn định hơn, dễ dàng mở rộng sang các hệ thống phần cứng khác nhau mà không cần thay đổi cấu trúc điều khiển. Đồng thời, đơn vị PU cũng giúp đơn giản hóa cấu hình điều khiển, khi tất cả các tín hiệu đều được đưa về cùng một hệ quy chiếu chuẩn hóa.



*Sơ đồ khối chuyển đổi ADC sang PU*

**4.4.1.2. Khối điều khiển chính**

Khối Control System là phần chính của vòng điều khiển dòng trong điều khiển BLDC. Mục tiêu chính là tạo ra giá trị độ rộng xung PWM (Duty) để điều chỉnh dòng điện cung cấp cho động cơ sao cho bám sát giá trị đặt (Idc\_ref\_PU) thông qua một bộ điều khiển PI và thuật toán chuyển mạch 6-steps dựa trên vị trí rotor.

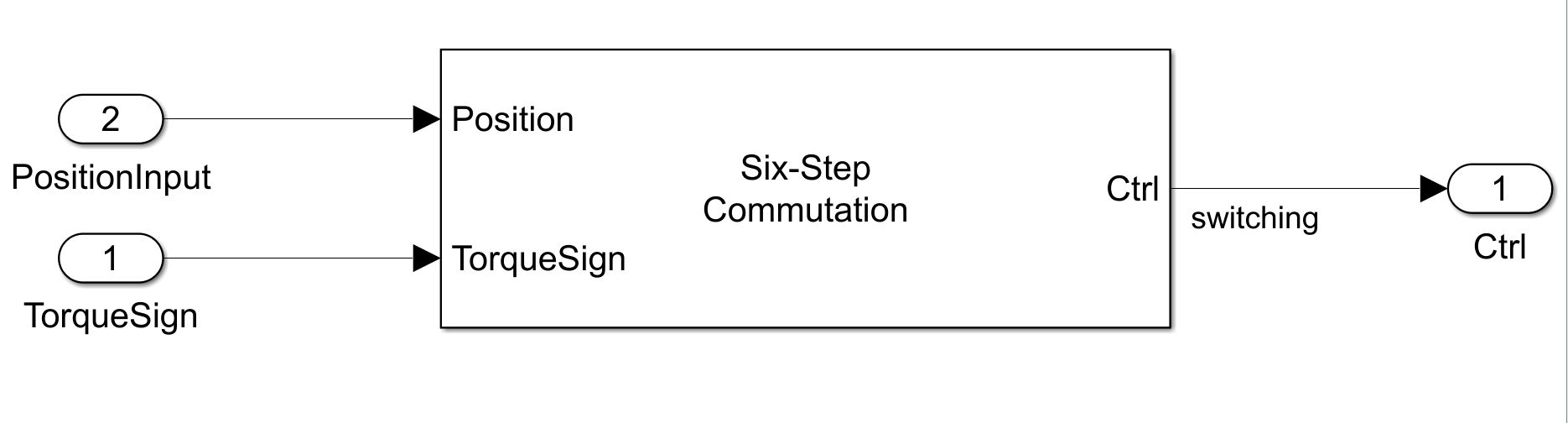


*Sơ đồ khối Control System*

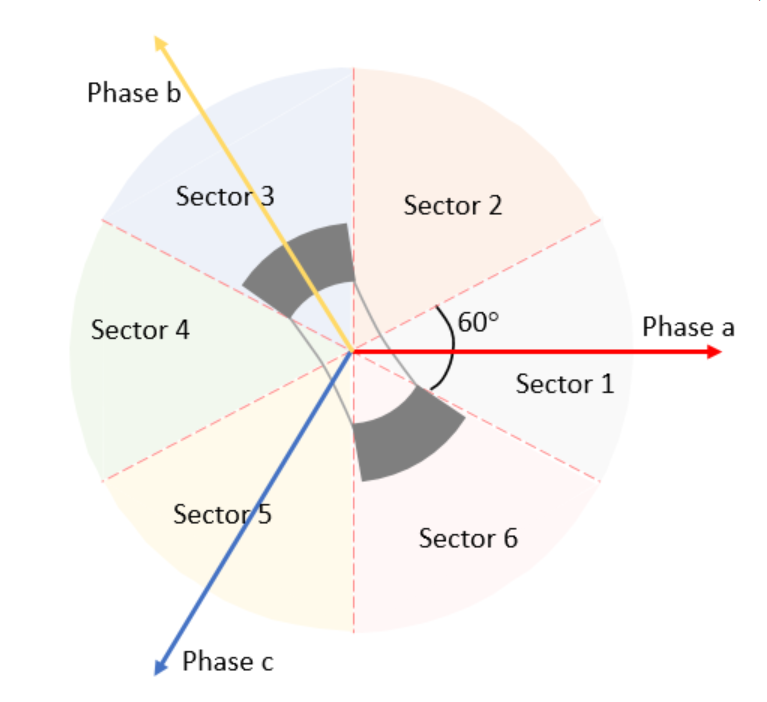
**4.4.1.2.1. Khối tạo tín hiệu chuyển mạch (Six Step Commutation)**

Khối Six-Step Commutation thực hiện chức năng chuyển mạch sáu bước (6-step switching) cho động cơ BLDC. Các cuộn dây của động cơ được kích hoạt theo một thứ tự nhất định gồm 6 trạng thái lặp lại, mỗi trạng thái ứng với 60 độ điện.

Khối này tính toán mẫu đóng/mở các MOSFET trong bộ nghịch lưu dựa trên vị trí điện của rotor và chiều quay, từ đó phát sinh tín hiệu điều khiển switching cho 3 pha U, V, W (hoặc A, B, C).



Dựa trên 2 tín hiệu đầu vào: góc điện (Position Input – chuyển đổi từ encoder) và chiều quay (Torque sign – lấy từ Idc\_ref gửi bởi host PC), khối sẽ xác định miền hiện tại của rotor. [5]



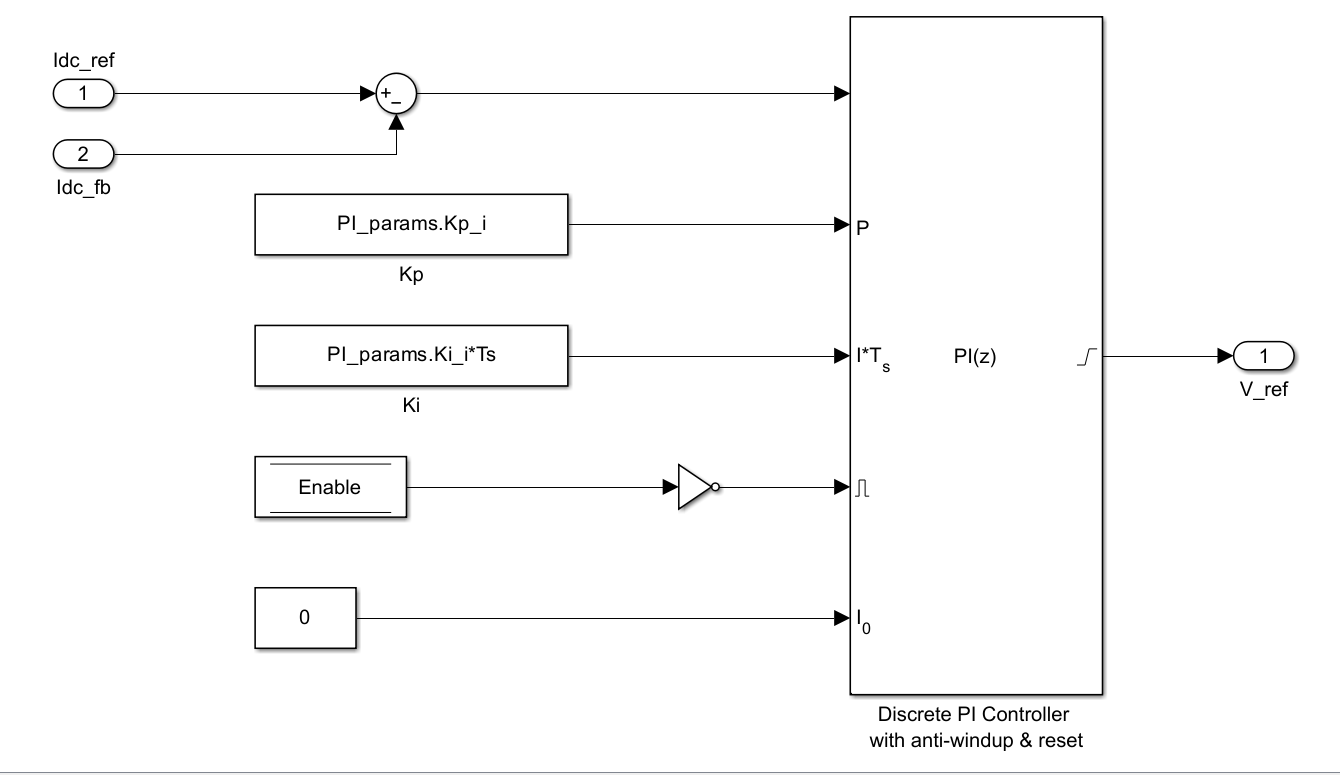
*Phân chia miền và các vector pha từ trường của stator*

Bảng tham chiếu miền và dòng điện pha tương ứng với góc rotor [5]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vị trí (θ)** | **Miền** | **Chuỗi chuyển đổi (AA' BB' CC')** | | |
| **AA'** | **BB'** | **CC'** |
| (-30°, 30°] | 1 | 00 | 10 | 01 |
| (30°, 90°] | 2 | 01 | 10 | 00 |
| (90°, 150°] | 3 | 01 | 00 | 10 |
| (150°, 210°] | 4 | 00 | 01 | 10 |
| (210°, 270°] | 5 | 10 | 01 | 00 |
| (270°, 330°] | 6 | 10 | 00 | 01 |

**4.4.1.2.2. Khối điều khiển PI dòng điện**

Khối điều khiển PI được sử dụng là bộ điều khiển dòng điện rời rạc, được thiết kế để đảm bảo dòng thực tế Idc\_fb bám sát theo dòng tham chiếu Idc\_ref, giúp duy trì moment quay mong muốn và đảm bảo tính ổn định cho hệ thống.



*Sơ đồ khối điều khiển dòng điện với PI*

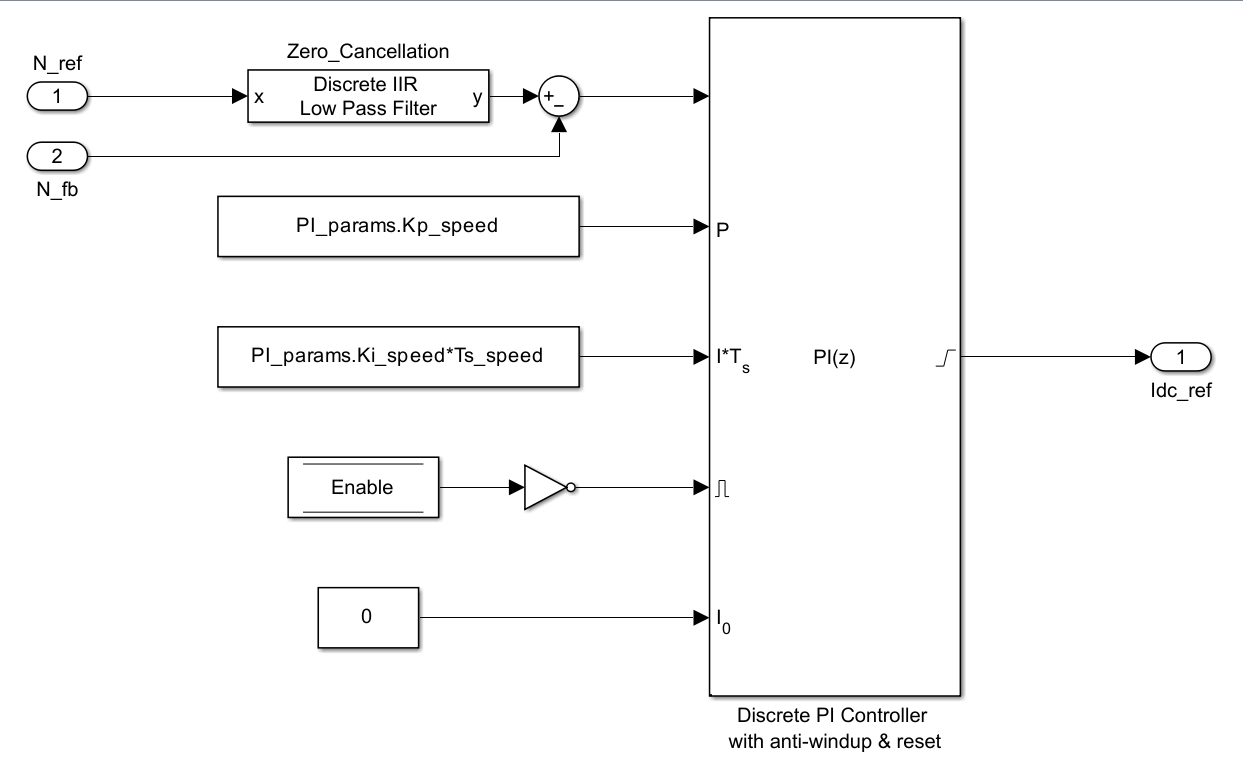
Thành phần P (Kp) được lấy từ tham số PI\_params.Kp\_i, tính toán dựa trên mô hình điện của động cơ, thành phần I (Ki) được đưa vào dưới dạng giá trị đã nhân với thời gian lấy mẫu Ts, tức là PI\_params.Ki\_i \* Ts, nhằm phù hợp với mô hình rời rạc hóa theo biến z. Các hệ số tỷ lệ này được tính bằng hàm tính toán của Matlab dựa trên thông số của động cơ, inverter và thời gian lấy mẫu mô phỏng bằng phương pháp Modulus Optimum (MO) [6].



Khối "Discrete PI Controller with anti-windup & reset" có cơ chế chống windup (chạy tích phân khi bộ điều khiển đang bị vô hiệu) tích hợp sẵn, giúp bộ điều khiển không bị mất ổn định khi đầu ra bị giới hạn (saturation) – một vấn đề thường gặp trong điều khiển dòng của inverter. Đầu ra của bộ điều khiển là V\_ref – giá trị điện áp tham chiếu điều chế, được nhân với tín hiệu chuyển mạch (tạo ra từ khối Six-step commutation) để tạo ra các xung PWM tương ứng với từng pha động cơ.

**4.4.2. Khối điều khiển vận tốc**

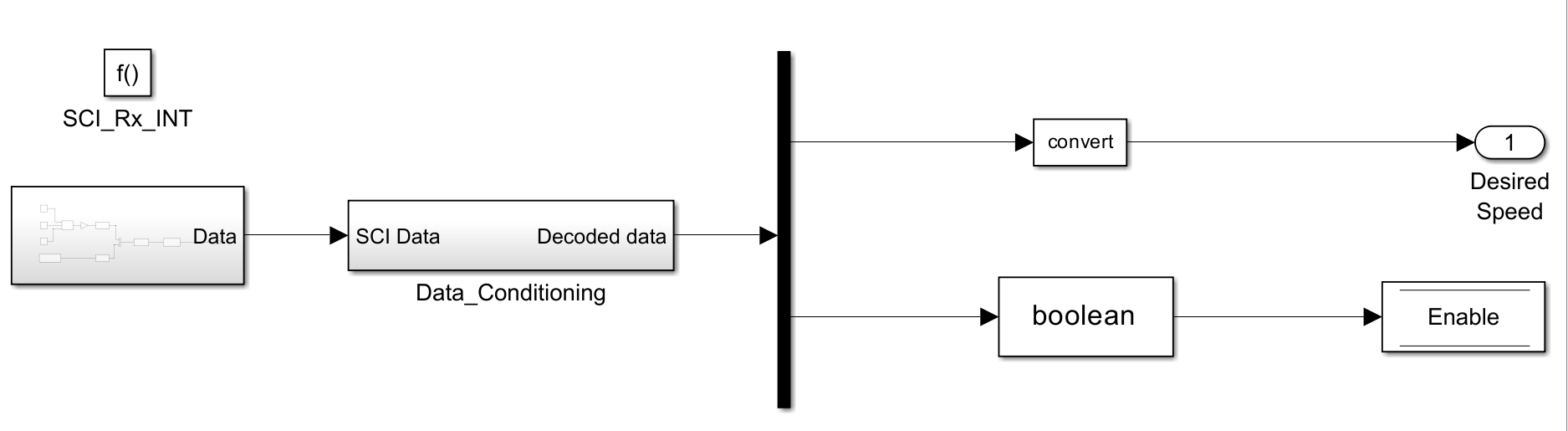
Khối điều khiển vận tốc là một vòng điều khiển ngoài (outer-loop), dùng để điều chỉnh tốc độ quay của động cơ BLDC thông qua việc điều khiển dòng điện tham chiếu (Idc\_ref) cấp cho vòng điều khiển dòng điện bên trong (inner-loop). Đây là một bộ điều khiển PI dạng rời rạc, thực hiện việc so sánh giữa tốc độ đặt và tốc độ phản hồi, sau đó sinh ra giá trị dòng điện phù hợp để điều khiển moment quay.



*Sơ đồ khối điều khiển PI vận tốc*

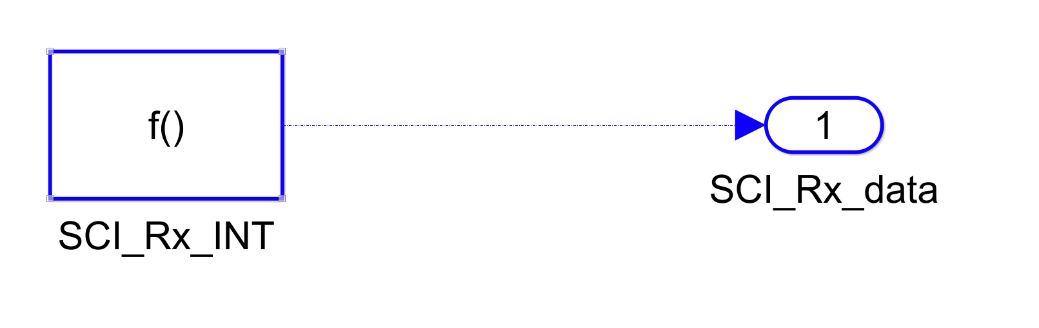
Thành phần P (Kp) và I (Ki) của bộ điều khiển tốc độ được lấy lần lượt từ các tham số PI\_params.Kp\_speed và PI\_params.Ki\_speed, trong đó thành phần I đã được nhân với thời gian lấy mẫu vòng tốc độ Ts\_speed, tức là PI\_params.Ki\_speed \* Ts\_speed, để phù hợp với mô hình rời rạc hóa theo biến z. Các hệ số này được tính toán tự động thông qua hàm nội bộ của MATLAB dựa trên thông số quán tính momen của động cơ, hệ thống chuẩn hóa PU, và chu kỳ lấy mẫu, theo phương pháp thiết kế Symmetrical Optimum (SO) [6] – một kỹ thuật điều chỉnh phổ biến trong điều khiển vòng ngoài của hệ truyền động điện nhằm đảm bảo khả năng đáp ứng tốt với nhiễu và quán tính cơ học lớn.

**4.4.3. Khối tạo tín hiệu giả lập UART**

****

Khối “Khối tạo tín hiệu giả lập UART” trong mô hình có vai trò mô phỏng việc nhận dữ liệu tốc độ và tín hiệu điều khiển từ máy tính hoặc thiết bị chủ (host) thông qua giao tiếp UART (Universal Asynchronous Receiver/Transmitter). Thay vì phải kết nối phần cứng vật lý và gửi dữ liệu thật, khối mô phỏng này giúp kiểm tra thuật toán điều khiển, luồng xử lý dữ liệu, và logic phản hồi tốc độ trong môi trường MATLAB/Simulink một cách nhanh chóng và linh hoạt. Việc mô phỏng tín hiệu UART như thế này là một phần thiết yếu trong quy trình Model-Based Design (MBD), giúp xác minh logic điều khiển trước khi tiến hành lập trình và nạp xuống vi điều khiển thực tế như F28069M.

**4.5. Khối host PC**



Trong mô hình, khối “SCI\_Rx\_INT” hoạt động như một phần host PC mô phỏng đóng vai trò phát lệnh định kỳ mỗi 0.1s đến hệ thống thông qua giao thức UART giả lập. Đây là khối Function-Call Generator, có nhiệm vụ tạo ra tín hiệu kích hoạt định kỳ để gọi thực thi khối con (subsystem) phía sau.

[1] “Surface Mount PMSM - Three-phase exterior permanent magnet synchronous motor with sinusoidal back electromotive force - Simulink.” Accessed: May 07, 2025. [Online]. Available: https://www.mathworks.com/help/mcb/ref/surfacemountpmsm.html

[2] “Converter (Three-Phase) - Controller-driven bidirectional AC/DC three-arm converter - MATLAB.” Accessed: May 07, 2025. [Online]. Available: https://www.mathworks.com/help/sps/ref/converterthreephase.html

[3] “Add Delay in Plant Model - MATLAB & Simulink.” Accessed: May 12, 2025. [Online]. Available: https://ww2.mathworks.cn/help/mcb/ug/add-delay-plantmoodeling.html?utm\_source=chatgpt.com

[4] “Create Motor Phase Current Sensing and Signal Conditioning Subsystem - MATLAB &amp; Simulink.” Accessed: May 12, 2025. [Online]. Available: https://ww2.mathworks.cn/help/mcb/ug/create-motor-phase-current-sensing-algorithm.html

[5] “Six Step Commutation - Generate switching sequence for six-step commutation of brushless DC (BLDC) motor - Simulink.” Accessed: May 12, 2025. [Online]. Available: https://www.mathworks.com/help/mcb/ref/sixstepcommutation.html

[6] “Estimate Control Gains and Use Utility Functions - MATLAB &amp; Simulink.” Accessed: May 13, 2025. [Online]. Available: https://www.mathworks.com/help/mcb/gs/estimate-control-gains-from-motor-parameters.html#mw\_49315eeb-a08d-4081-95f1-34e91365ef10