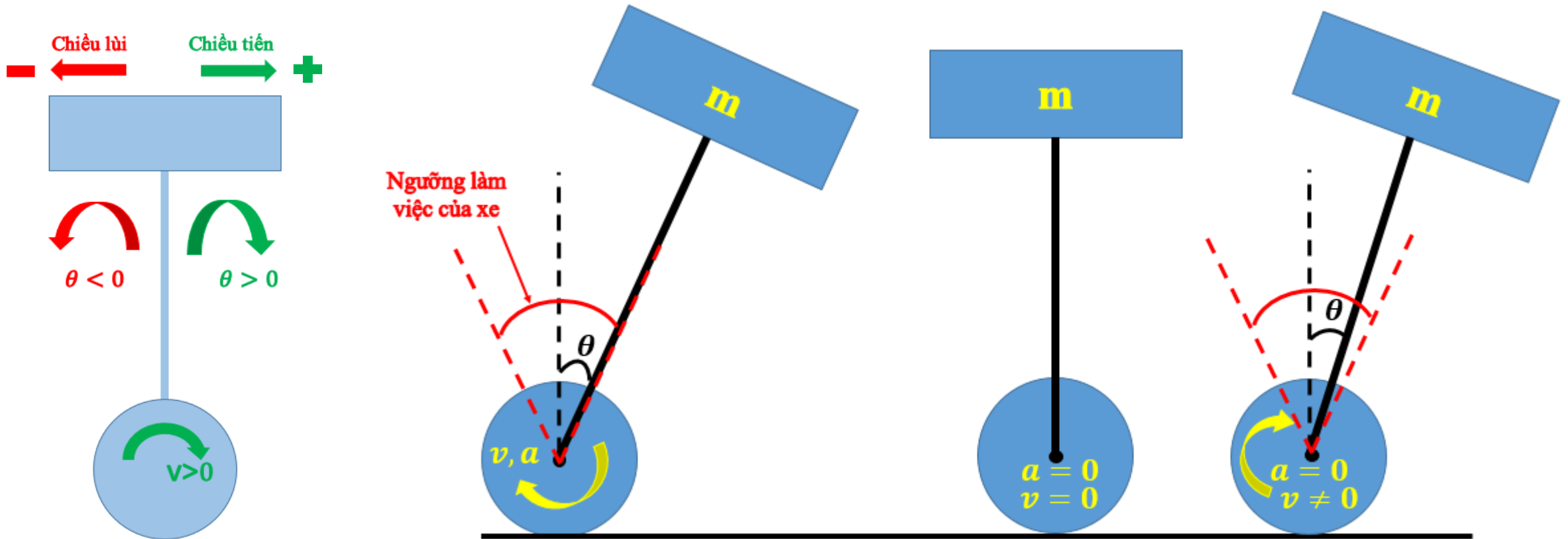


XE HAI BÁNH TỰ CÂN BẰNG

Tập Huấn SRobot 2018 - 2019
Saigon Institute of Technology



1. Nguyên Lý Hoạt Động

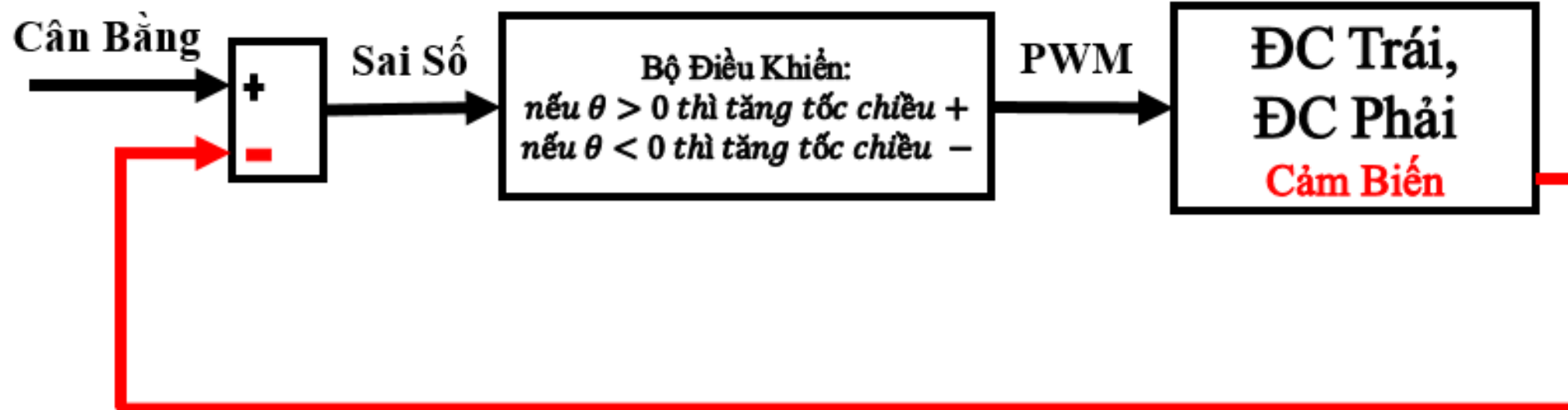


➤ Để xe cân bằng:

- Khi xe nghiêng về phía trước → tăng tốc về phía trước
- Khi xe nghiêng về phía sau → tăng tốc về phía sau
- Sai số góc nghiêng (so với góc cân bằng) càng lớn thì tăng tốc càng nhanh.

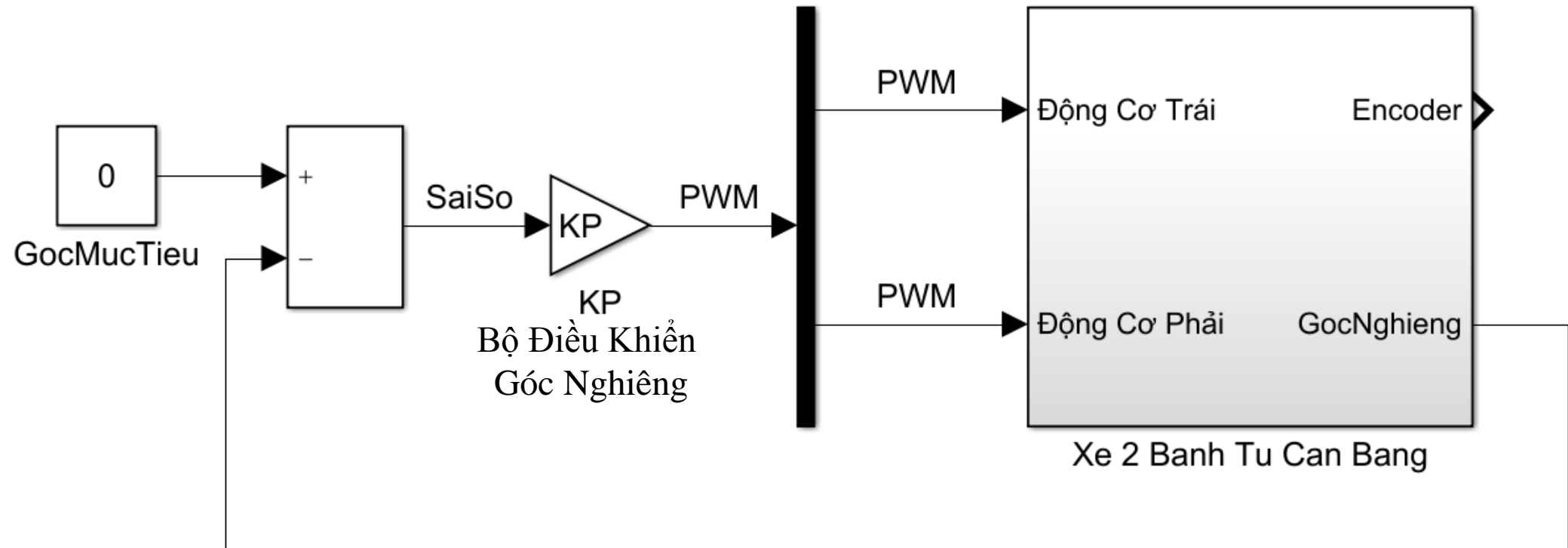
➤ Như vậy, chương trình sẽ điều khiển xe cho đến khi xe cân bằng ($a = 0$ và θ trong ngưỡng làm việc)

2. Hệ Thống Điều Khiển Hồi Tiếp



3. Mô Hình Cân Bằng

Bước 1: Cân bằng bằng cách làm giảm sai số góc nghiêng của xe



$$SaiSo = GocMucTieu - GocNghieng$$

$$PWM = KP * SaiSo \quad (\text{với } KP \text{ là hệ số khuếch đại})$$

Xe 2 bánh tự cân bằng, giả sử ta chọn được **KP=10**.
Giới hạn $PWM \in [-255, 255]$, $GocMucTieu = 0$

□ Ví dụ 1: Góc nghiêng của xe đang là $+10^0$
 $PWM = 10 * (0 - 10^0) = -100$

➤ Điều khiển xe đi tới với $PWM = 100$

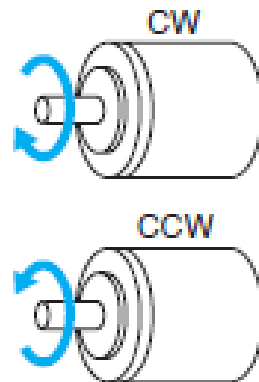
□ Ví dụ 2: Góc nghiêng của xe đang là -7^0
 $PWM = 10 * (0 - (-7^0)) = 70$

➤ Điều khiển xe đi lùi với $PWM = 70$

□ Ví dụ 3: Góc nghiêng của xe đang là -30^0
 $PWM = 10 * (0 - (-30^0)) = 300$

➤ Vì $300 > 255 \rightarrow PWM = 255$

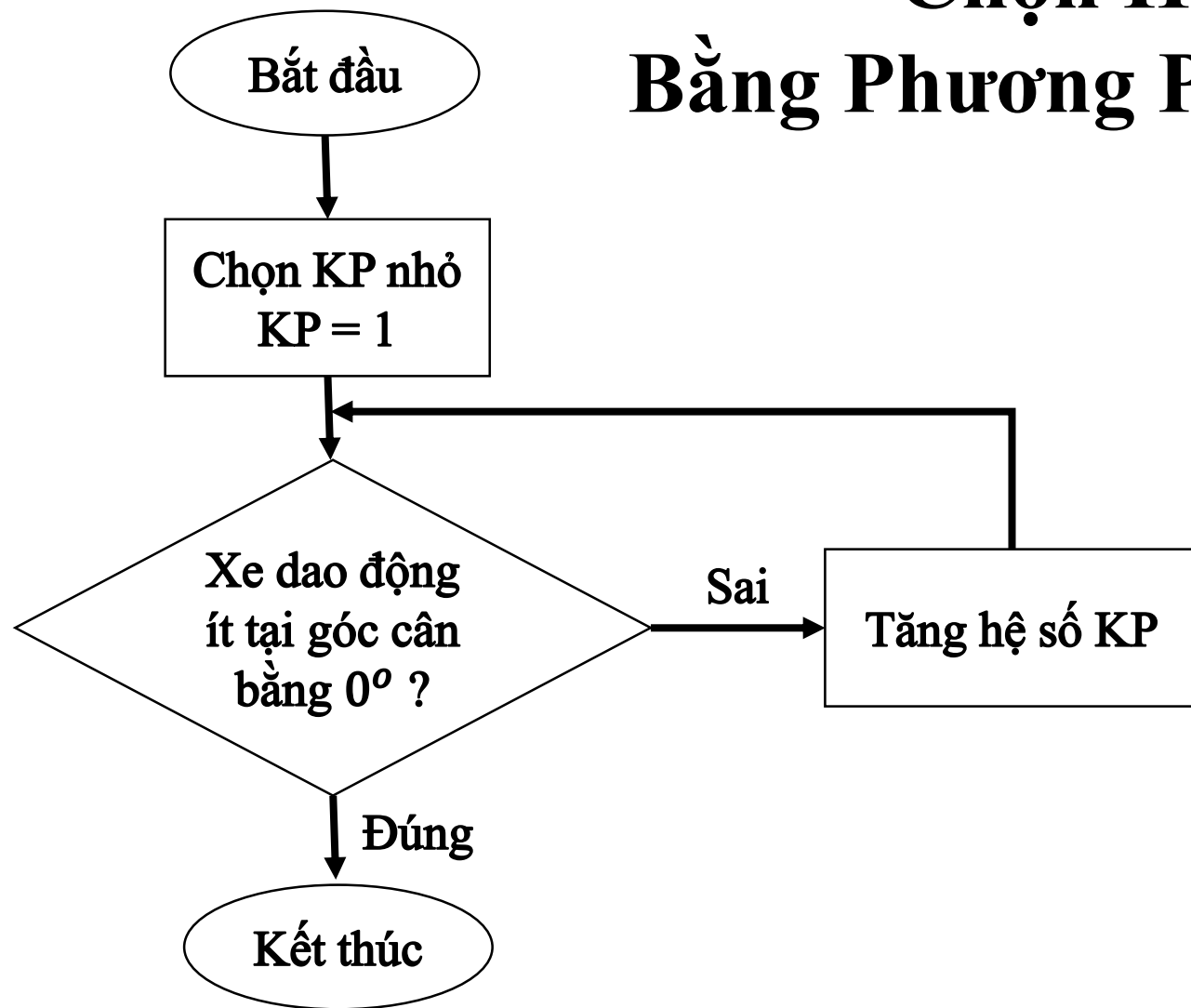
➤ Điều khiển xe đi lùi với $PWM = 255$



Chọn KP = 40	
Góc Nghiêng	PWM
12.78^0	255, đi tới
5.30^0	212, đi tới
-10.33^0	255, đi lùi
-2.50^0	100, đi lùi
0^0	0, xe đứng im
4.5^0	180, đi tới

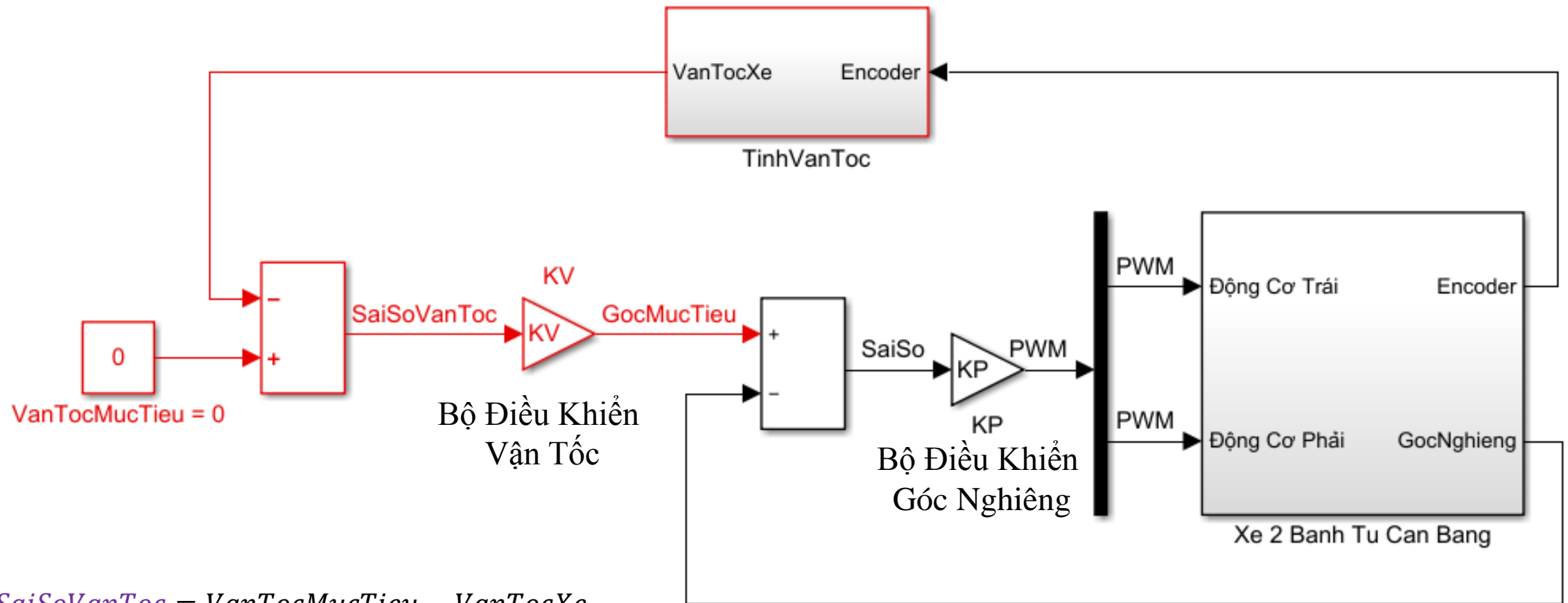
Chọn Hệ Số KP

Bằng Phương Pháp Thử - Sai



- Nếu chọn **KP** quá nhỏ, làm pwm sẽ nhỏ → điện áp cấp vào động cơ nhỏ → động cơ quay nhẹ → không đủ lực để xe cân bằng → xe ngã theo chiều đang nghiêng.
- Nếu chọn **KP** quá lớn, dù xe chỉ mới nghiêng ít của chiều bên này, nhưng KP quá lớn làm pwm lớn → điện áp cấp vào động cơ lớn → động cơ quay mạnh → khiến xe bật mạnh sang chiều bên kia

Bước 2: Cải thiện sự cân bằng bằng cách làm giảm sai số vận tốc



$$SaiSoVanToc = VanTocMucTieu - VanTocXe$$

$$GocMucTieu = KV * SaiSoVanToc$$

$$SaiSo = GocMucTieu - GocNghieng$$

$$PWM = KP * SaiSo$$

Xe 2 bánh tự cân bằng, giả sử ta chọn được $KP = 10$, $KV = 0.05$, xe đứng yên ($VanTocMucTieu = 0$). Giới hạn góc mục tiêu $GocMucTieu \in [-20^\circ, 20^\circ]$, giới hạn $PWM \in [-255, 255]$.

□ Ví dụ 1: Góc nghiêng của xe đang là $+10^\circ$, $v_{tt} = +50v/p$

$$\text{SaiSoVT} = 0 - 50 = -50. \quad \text{GocMucTieu} = 0.05 * (-50) = -2.5^\circ \quad \text{SaiSo} = -2.5^\circ - 10^\circ = -12.5^\circ$$

$$PWM = 10 * (-12.5) = -125$$

➤ Điều khiển xe đi tới với $PWM = 125$

□ Ví dụ 2: Góc nghiêng của xe đang là $+10^\circ$, $v_{tt} = +100v/p$

$$\text{SaiSoVT} = 0 - 100 = -100 \quad \text{GocMucTieu} = 0.05 * (-100) = -5^\circ \quad \text{SaiSo} = -5^\circ - 10^\circ = -15^\circ$$

$$PWM = 10 * (-15) = -150$$

➤ Điều khiển xe đi tới với $PWM = 150$

□ Ví dụ 3: Góc nghiêng của xe đang là -20° , $v_{tt} = -250v/p$

$$\text{SaiSoVT} = 0 - (-250) = 250 \quad \text{GocMucTieu} = 0.05 * 250 = 12.5^\circ \quad \text{SaiSo} = 12.5^\circ - (-20^\circ) = 32.5^\circ$$

$$PWM = 10 * 32.5 = 325$$

➤ Ta thấy $325 > 255$ nên giá trị PWM lớn nhất là 255.

➤ Điều khiển xe đi lui với $PWM = 255$

KV là hệ số giúp xe cân bằng nhanh hơn dựa vào vận tốc của bánh xe.

- Ta đang muốn xe cân bằng do đó đặt $VanTocMucTieu = 0$.
- Khi cần xe chạy với một vận tốc thì cho $VanTocDat \neq 0$, ví dụ $VanTocMucTieu = +50$ vòng/phút

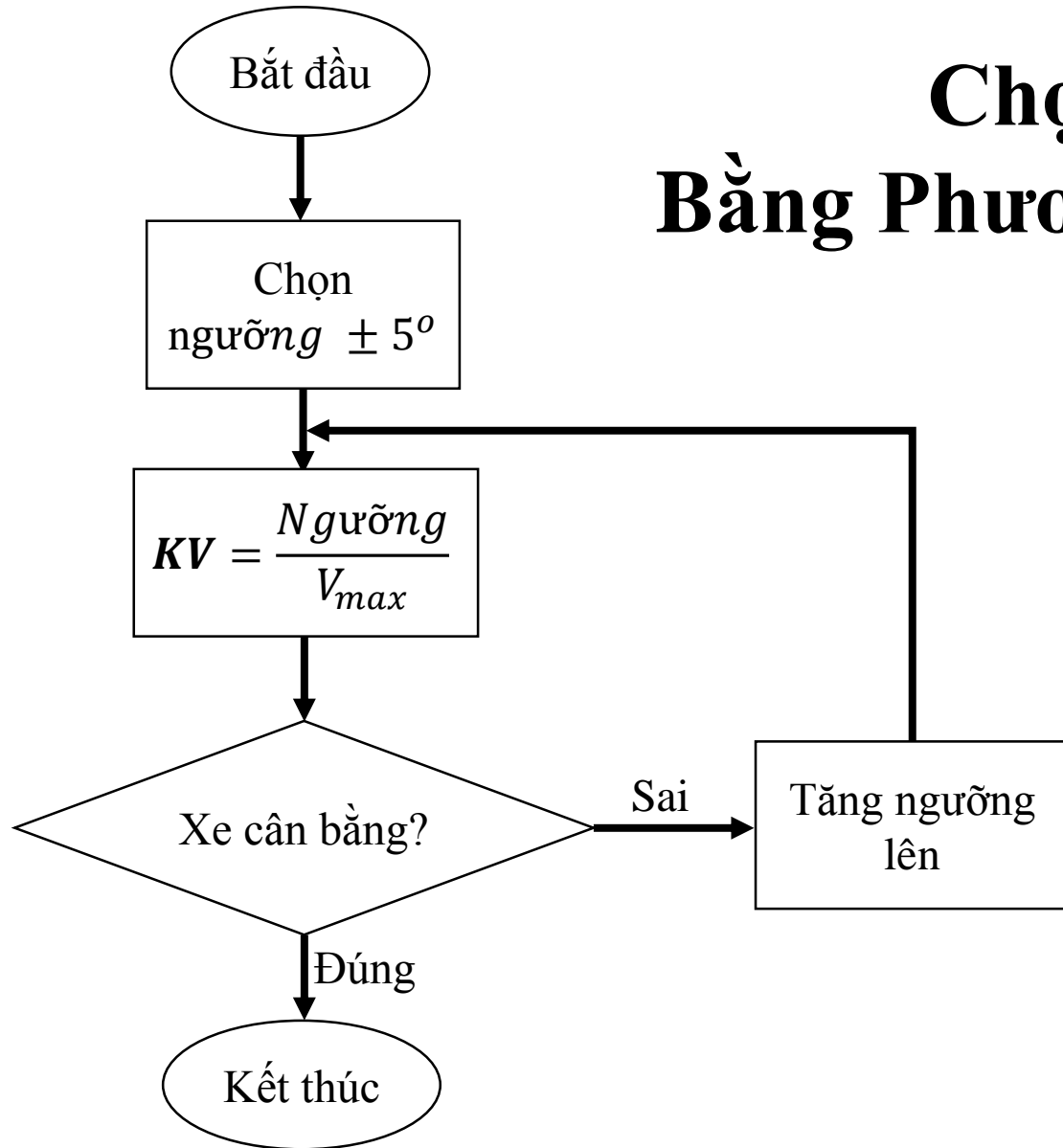
$$KV = \frac{\text{Ngưỡng làm việc của xe}}{\text{Vận tốc tức thời lớn nhất}}$$

Giả sử ta cho động cơ quay với tốc độ tối đa

`analogWrite(PinPWM, 255)`, và đo được vận tốc bánh xe (quay không tải) là 400 v/p. Ta thử - sai lần lượt các trường hợp sau

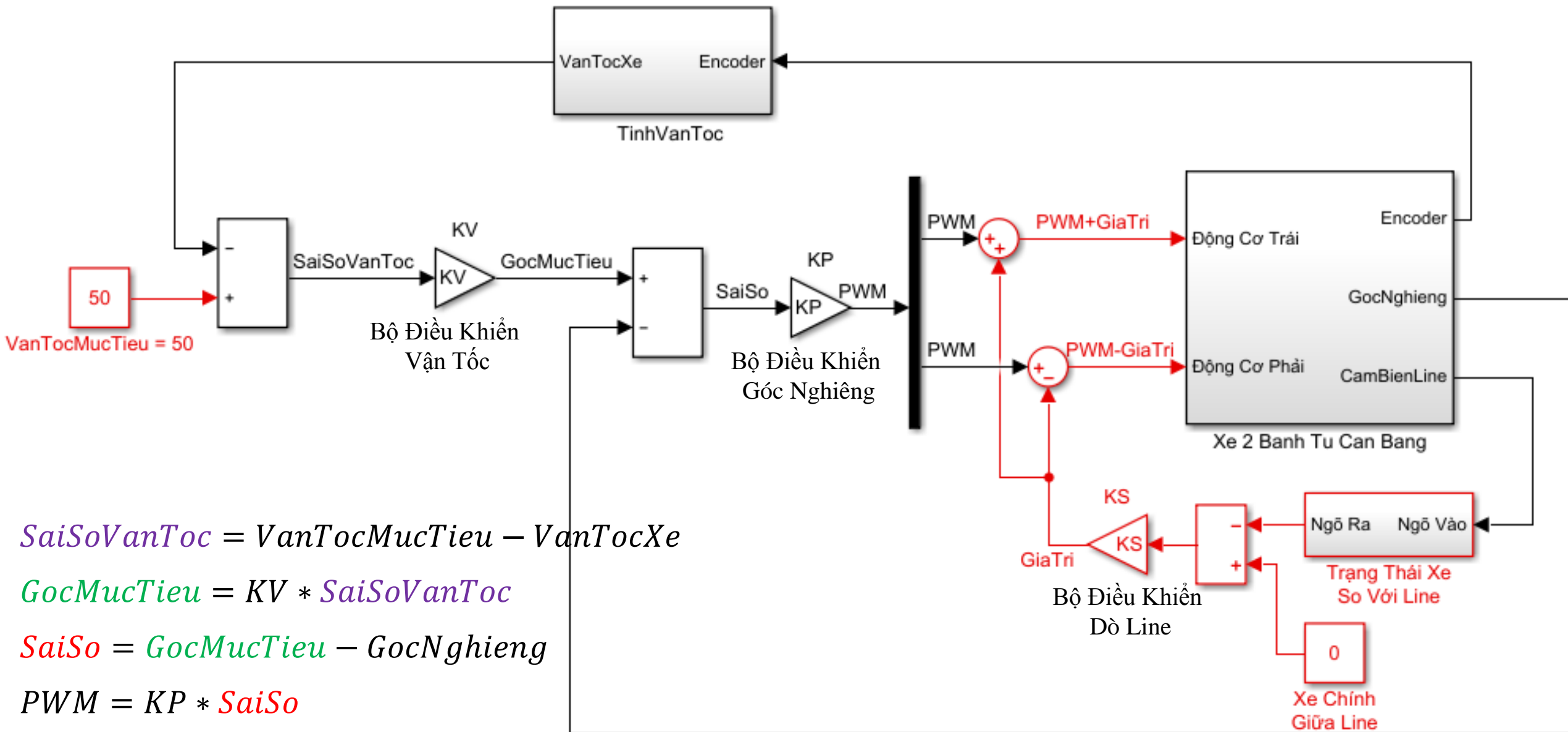
Ngưỡng làm việc của xe	$\pm 5^\circ$	$\pm 10^\circ$	$\pm 12^\circ$	$\pm 15^\circ$	$\pm 18^\circ$	$\pm 20^\circ$	$\pm 25^\circ$
Chọn KV > 0	0.0125	0.025	0.03	0.0375	0.045	0.05	0.0625

Chọn Hệ Số KV Bằng Phương Pháp Thử - Sai



4. Dò Line

- Dùng cảm biến dò line (5 cặp led) để xác định xe lệch bên trái hay bên phải đường đi
- Khi **xe lệch** về bên **trái** của đường đi, **điều khiển** xe đi về bên **phải**: Vận tốc bánh trái lớn hơn vận tốc bánh phải
- Khi **xe lệch** về bên **phải** của đường đi, **điều khiển** xe đi về bên **trái**: Vận tốc bánh phải lớn hơn vận tốc bánh trái



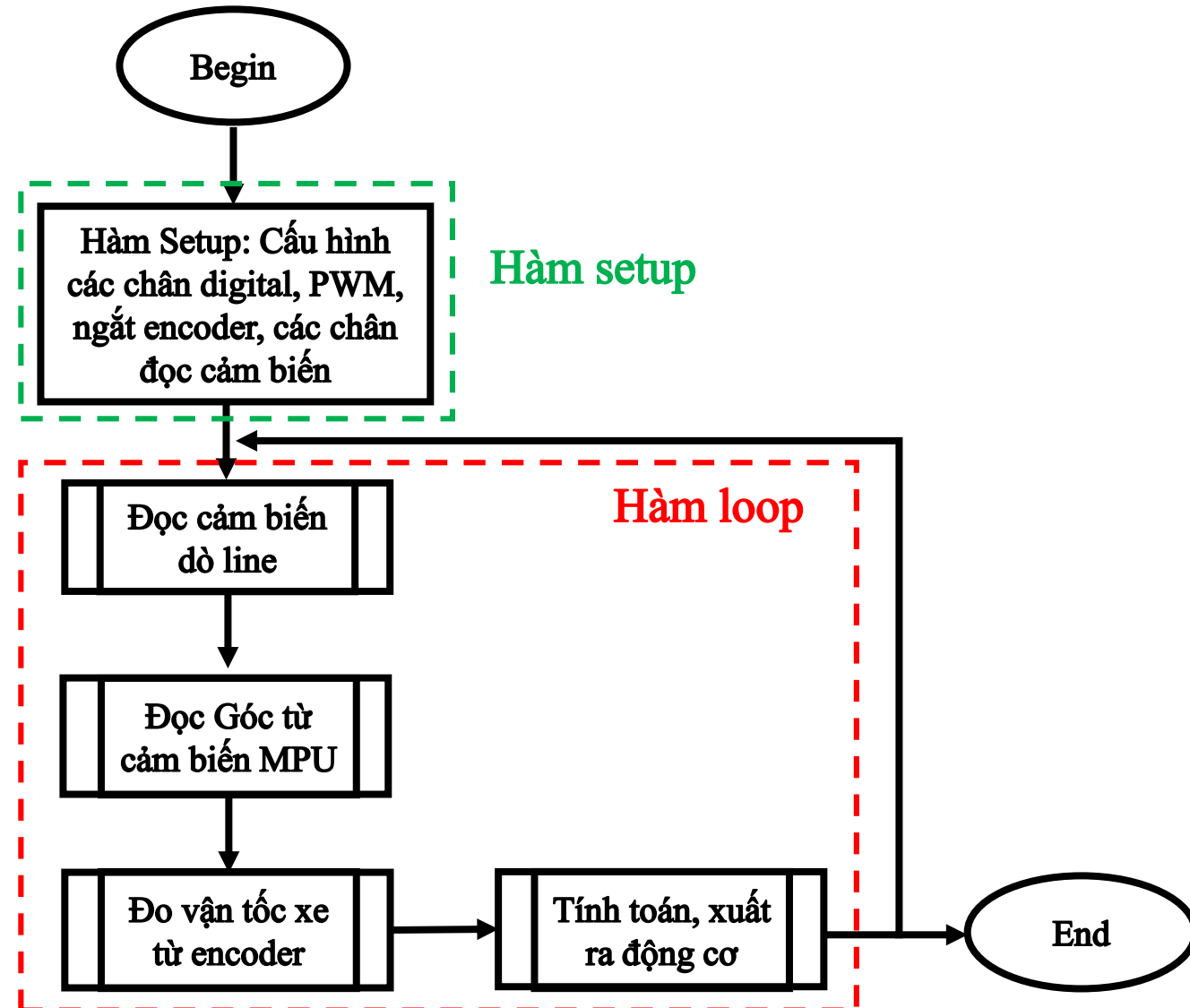
Bảng Giá Trị

STT	Đọc từ cảm biến dò line	Giá trị sai số của xe so với đường line	Ý nghĩa
1	01111	4	Đường line lệch rất nhiều phía bên trái của xe
2	00111	3	Đường line lệch nhiều phía bên trái của xe
3	10111	2	Đường line lệch ít phía bên trái của xe
4	10011	1	Đường line lệch rất ít phía bên trái của xe
5	11011	0	Đường line nằm chính giữa của xe
6	11001	-1	Đường line lệch rất ít phía bên phải của xe
7	11101	-2	Đường line lệch ít phía bên phải của xe
8	11100	-3	Đường line lệch nhiều phía bên phải của xe
9	11110	-4	Đường line lệch rất nhiều phía bên phải của xe
10	11111	+5 hoặc -5	Xe ra khỏi đường line, về phía trái hoặc phải

Các giá trị trên đây chỉ mang tính minh họa, các bạn có thể tùy chọn giá trị khác miễn là chúng phải đối xứng nhau qua 0.

5. Hướng Dẫn Thực Hiện

Lưu
Đồ
Giải
Thuật



Bước 1: Lắp ráp khung xe, kết nối dây điện vào các thiết bị.

Bước 2: Quy ước chiều của xe

Bước 3: Viết chương trình cân bằng ($VanTocMucTieu = 0$)

Tìm hệ số KP trước. ($KV=0$)

Tìm hệ số KV sau (giữ nguyên hệ số KP đã tìm trước đó)

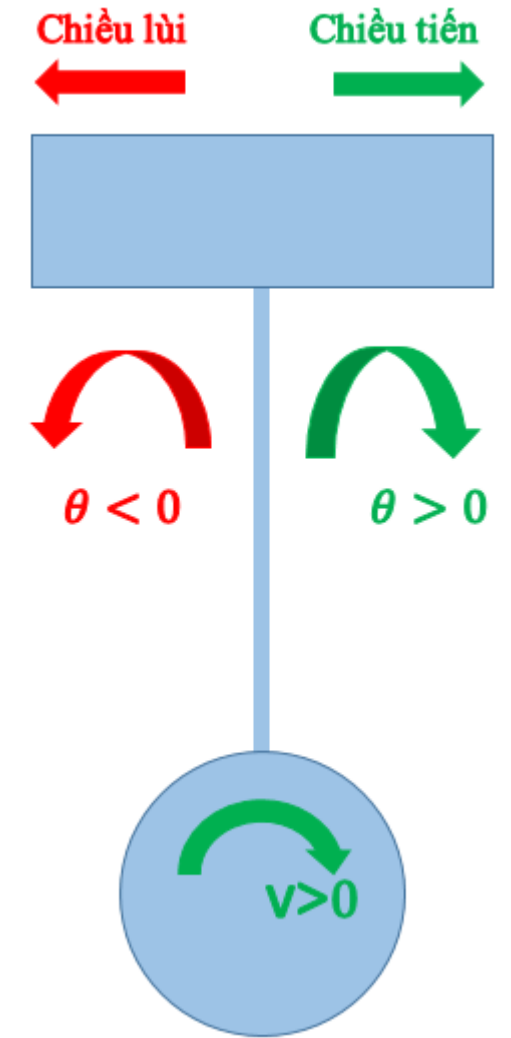
Bước 4: Lắp thanh cảm biến dò line vào xe.

Khi lắp thanh dò line vào, xe của ta sẽ có xu hướng nghiêng về phía mà thanh dò line mới được lắp, do đó, ta sẽ offset góc nghiêng như sau:

$GocNghieng = GocNghiengTuMPU + OFFSET; //$ OFFSET có thể âm hoặc dương.

Bước 5: Viết chương trình dò line. ($VanTocMucTieu \neq 0$)

Bước 6: Chạy thử và hiệu chỉnh lại KP, KV, offset,... (nếu cần)



Quy ước chiều của xe tại bước 2

Pseudo code: Dò line

```
float DoLine()          // chương trình con của hàm dò line.
{
    // các trường hợp ta xem từ Bảng Giá Trị tại slide trang 13
    đọc cảm biến từ thanh dò line;
    if      ( trường hợp 1)      { SaiSo = 4; RaNgoai = 1;    }
    else if ( trường hợp 2)      { SaiSo = 3; RaNgoai = 1;    }
    else if ( trường hợp 3)      { SaiSo = 2;                }
    else if ( trường hợp 4)      { SaiSo = 1;                }
    else if ( trường hợp 5)      { SaiSo = 0;                }
    else if ( trường hợp 6)      { SaiSo = -1;               }
    else if ( trường hợp 7)      { SaiSo = -2;               }
    else if ( trường hợp 8)      { SaiSo = -3; RaNgoai = -1; }
    else if ( trường hợp 9)      { SaiSo = -4; RaNgoai = -1; }
    else if ( xe ra khỏi line )  // trường hợp 11
    {
        if ( RaNgoai == 1 ) { SaiSo = 5;    } // line lệch trái của xe
        else                { SaiSo = -5;   } // line lệch phải của xe
    }
    GiaTri = KS * SaiSo;        // tìm KS theo phương pháp thử - sai
    return GiaTri;
}
```


Pseudo Code: Tính toán xuất ra động cơ

```
void TinhToanXuatRaDC(float VanTocXe, float GiaTriDoLine, float GocNghiemg)
{
    VanTocMucTieu = 40; // vì xe dò line, nên phải đặt một vận tốc khác 0, ở đây vận tốc đặt là 40 vòng/phút
    SaiSoVanToc = VanTocMucTieu - VanTocXe;
    GocMucTieu = KV * SaiSoVanToc;
    SaiSo = GocMucTieu - GocNghiemg;
    Pwm = KP * SaiSo;
    PwmTrai = Pwm + GiaTriDoLine ;
    PwmPhai = Pwm - GiaTriDoLine ;
    if (PwmTrai > 0)
    {
        if (PwmTrai > 255)    PwmTrai = 255;
        Thiết lập động cơ trái quay lui;
    }
    else // trường hợp PwmTrai < 0
    {
        if (PwmTrai < -255)    PwmTrai = -255;
        Thiết lập động cơ trái quay tới;
        PwmTrai = abs(PwmTrai ); // lấy trị tuyệt đối, đảm bảo PwmTrai không âm
    }
}
```

```
1 → if (PwmPhai > 0)
{
    if (PwmPhai > 255)    PwmPhai = 255;
    Thiết lập động cơ phải quay lui;
}
else // trường hợp PwmPhai < 0
{
    if (PwmPhai < -255)    PwmPhai = -255;
    Thiết lập động cơ phải quay tới;
    PwmPhai = abs(PwmPhai);
}
Xuất giá trị PwmTrai ra động cơ trái ;
Xuất giá trị PwmPhai ra động cơ phải ;
} // kết thúc hàm
```