

Clock 0-5:

```
S_k1[i] = HsVd2_VECT6[i] - iHsVm_VECT6[i]
a1_m_Vd_d1[i] = iAnew[0] × HsVd1_VECT6[i]
a2_m_Vd_d2[i] = iAnew[1] × HsVd2_VECT6[i]
lamc_M_S1_k1_esti[i] = S1_k1_esti[i] × ilambda
```

Clock 9-14:

```
Output_diff[i] = S_k1[i] - S1_k1_esti[i]
```

Clock 15-23:

```
Vd_s_a1Vd1[i] = iHsVd_VECT6[i] - a1_m_Vd_d1[i]
Vd_s_a1Vd1_s_a2Vd2[i] = Vd_s_a1Vd1[i] - a2_m_Vd_d2[i]
```

```
mVolt_6x1_3 = Output_diff
mTransf_MAT6x6_3 = i_A1matrix_s_lambda
```

```
mVolt_6x1_2 = Output_diff
mTransf_MAT6x6_2 = i_L2_esti
```

Clock 18-23:

```
S1_esti_correct[i] = L1 × Output_diff[i] // L1 = iAnew[2]
```

Clock 24-32:

```
S2_esti_correct[i] = L2 × Output_diff[i] // L2 = iAnew[3]
```

Clock 25-30:

```
S1_esti[i] = lamc_M_S1_k1_esti[i] + S1_esti_correct[i]
```

Clock 34-39:

```
S2_esti[i] = S1_k1_esti[i] + S2_esti_correct[i]
a1lamc_m_S1_esti[i] = a1_s_lamc × S1_esti[i] // a1_s_lamc =
iAnew[4]
```

Clock 41-46:

```

Vd_s_a1Vd_s_a2Vd2_p_allamc_m_S1_esti[i] =
    Vd_s_a1Vd1_s_a2Vd2[i] + allamc_m_S1_esti[i]
a2_m_S2_esti[i] = iAnew[1] × S2_esti[i]

```

Clock 49-54:

```

u_before_Binv[i] = Vd_s_a1Vd_s_a2Vd2_p_allamc_m_S1_esti[i] +
a2_m_S2_esti[i]

```

Clock 56-67:

```

one_p_Betam_m_westi_d1[i] = one_p_Betam × westi_d1[i]

```

```

Betam_m_w2esti_d1[i] = iBeta_m × w2esti_d1[i]

```

```

w1_before_correct[i] = westi_d1[i] + w2esti_d1[i]

```

Clock 64:

```

mVolt_6x1_4 = u_before_Binv
mTransf_MAT6x6_4 = i_inv_B          //  $B^{-1}$ 

```

Clock 92-93

```

w_correct = Rst_6x6_6x1_3
w2_correct = Rst_6x6_6x1_2

```

Clock 101-118:

```

westi[i] = w1_before_correct[i] + w_correct[i]

```

```

w2esti[i] = w2esti_d1[i] + w2_correct[i]

```

Clock 133:

```

u_before_w = Rst_6x6_6x1_4

```

Clock 134-147:

```

u_total_new[i] = u_before_w[i] - westi[i]

```

```
westi_d1[i] ≤= westi[i]  
w2esti_d1[i] ≤= w2esti[i]  
S1_k1_esti[i] ≤= S1_esti[i]  
S2_k1_esti[i] ≤= S2_esti[i]  
HsVd2_VECT6[i] ≤= HsVd1_VECT6[i]  
HsVd1_VECT6[i] ≤= iHsVd_VECT6[i]
```

FPGA 控制器核心數學運算表達式

1. 基本控制變數定義

狀態變數:

- S_k : 追蹤誤差 (**tracking error**)
- S_1^{esti} : 狀態估測器輸出
- S_2^{esti} : 延遲狀態估測器
- w^{esti} : 擾動估測
- w_2^{esti} : 二階擾動估測

2. 主要控制律運算

Step 1: 追蹤誤差計算

$$S_k = V_d - V_m$$

Step 2: 前饋控制

$$V_{\text{ff}} = V_d - a_1 \times V_{d-1} - a_2 \times V_{d-2}$$

Step 3: 狀態估測誤差

$$e = S_k - S_1^{\text{esti}_{k-1}}$$

Step 4: 估測器更新

$$S_1^{\text{esti}} = \lambda_c \times S_1^{\text{esti}_{k-1}} + L_1 \times e$$

$$S_2^{\text{esti}} = S_1^{\text{esti}_{k-1}} + L_2 \times e$$

Step 5: 回饋控制

$$u_{\text{fb}} = (a_1 - \lambda_c) \times S_1^{\text{esti}} + a_2 \times S_2^{\text{esti}}$$

Step 6: 控制輸入(未解耦)

$$u_{\text{before}} = V_{\text{ff}} + u_{\text{fb}}$$

Step 7: 矩陣解耦

$$u_{\text{decoupled}} = B^{-1} \times u_{\text{before}}$$

Step 8: 擾動補償

$$u_{\text{final}} = u_{\text{decoupled}} - w^{\text{esti}}$$

3. 擾動估測器運算

擾動估測更新:

$$w_1^{\text{before}} = w^{\text{esti}_{k-1}} + w_2^{\text{esti}_{k-1}}$$

$$w^{\text{correct}} = (\text{A1matrix_s_lambda}) \times e$$

$$w_2^{\text{correct}} = (\text{L2_esti}) \times e$$

$$w^{\text{esti}} = w_1^{\text{before}} + w^{\text{correct}}$$

$$w_2^{\text{esti}} = w_2^{\text{esti}_{k-1}} + w_2^{\text{correct}}$$

增強版本 (with β_m):

$$w_1^{\text{before}} = (1 + \beta_m) \times w^{\text{esti}_{k-1}} + \beta_m \times w_2^{\text{esti}_{k-1}}$$

4. 關鍵系統參數

固定參數:

- $a_1 = 1.595052025060797$ (來自系統極點 p_1)
- $a_2 = -0.599079946700523$ (來自系統極點 p_2)
- $\lambda_c = 0.9391$ (控制頻寬 **1000 Hz**)
- $\lambda_w = 0.7304$ (估測器頻寬 **5000 Hz**)
- β_m : 模型參數 (可調)

5. 估測器增益計算

Augmented Version (當前活躍版本):

$$L_1 = - \left(-\beta + (1 + \beta)^2 - 4(1 + \beta)\lambda_w + 6\lambda_w^2 - \frac{\lambda_w^4}{\beta} \right)$$

$$L_2 = - \left(1 + \beta - 4\lambda_w - (1 + \beta) \frac{\lambda_w^4}{\beta^2} + 4 \frac{\lambda_w^3}{\beta} \right)$$

$$L_3 = \text{tune_L3}$$

(來自 **ThreeD_sine_theta**)

iAnew 參數映射:

- **iAnew[0] = $a_1 = 1.595052025060797$**
- **iAnew[1] = $a_2 = -0.599079946700523$**
- **iAnew[2] = L_1 (單一增益值)**
- **iAnew[3] = L_2 (單一增益值)**
- **iAnew[4] = $a_1 - \lambda_c$**

6. 矩陣運算

解耦矩陣 **B⁽⁻¹⁾**:

$$B^{-1} = \begin{pmatrix} 1254.4 & 365.9 & 362.7 & 509.3 & 510.7 & 478.1 \\ 354.7 & 1162.4 & 387.9 & 669.9 & 692.5 & 448.6 \\ 427.5 & 453.9 & 1258.9 & 439.3 & 488.0 & 520.8 \\ 442.5 & 654.6 & 353.8 & 1460.7 & 800.5 & 448.8 \\ 414.2 & 572.5 & 337.4 & 687.5 & 1274.4 & 374.7 \\ 412.2 & 400.9 & 368.5 & 474.4 & 407.4 & 1371.6 \end{pmatrix} \times L_3$$

擾動估測矩陣:

$$\text{Almatrix_s_lambda} = B^{-1} \times L_1$$

$$\text{L2_esti} = B^{-1} \times L_2$$

7. 完整控制輸入計算流程 (Control Effort Computation)

Step 1: 追蹤誤差計算

$$\delta v[k] = v_d[k] - v_m[k]$$

Step 2: 前饋控制項 (Feedforward)

$$v_{\text{ff}}[k] = v_d[k + 1] - a_1 v_d[k] - a_2 v_d[k - 1]$$

Step 3: 估測追蹤誤差 (使用增強狀態估測器)

$$\widehat{\delta \mathbf{v}}[k] = \hat{\mathbf{s}}_1[k]$$

Step 4: 回饋控制項 (Feedback)

$$\delta \mathbf{v}_{\text{fb}}[k] = (a_1 - \lambda_c) \widehat{\delta \mathbf{v}}[k] + a_2 \widehat{\delta \mathbf{v}}[k-1]$$

Step 5: 總控制輸入 (未解耦)

$$\mathbf{v}_{\text{total}}[k] = \mathbf{v}_{\text{ff}}[k] + \delta \mathbf{v}_{\text{fb}}[k]$$

Step 6: 矩陣解耦與擾動補償

$$\mathbf{u}[k] = \mathbf{B}^{-1} \mathbf{v}_{\text{total}}[k] - \hat{\mathbf{w}}[k]$$

其中 \mathbf{B}^{-1} 為預先計算的解耦矩陣， $\hat{\mathbf{w}}[k]$ 來自擾動估測器

Step 7: 最終控制輸出 (Control Effort)

$$\mathbf{u}_{\text{final}}[k] = \mathbf{B}^{-1} [\mathbf{v}_d[k+1] - a_1 \mathbf{v}_d[k] - a_2 \mathbf{v}_d[k-1] + (a_1 - \lambda_c) \widehat{\delta \mathbf{v}}[k] + a_2 \widehat{\delta \mathbf{v}}[k-1]] - \hat{\mathbf{w}}[k]$$

此控制輸入將產生磁通：

$$\boldsymbol{\varphi}[k] = \mathbf{B} \mathbf{u}_{\text{final}}[k]$$

進而產生磁力：

$$\mathbf{f}_m = g_{\varphi} \boldsymbol{\varphi}^T \mathbf{L}(\mathbf{p}) \boldsymbol{\varphi}$$

iAnew

```
Anew[0][0] = A1matrix[0][0] = 1.595052025060797
Anew[0][1] = A2matrix[0][0]=-0.599079946700523;
Anew[0][2] = 1 + beta_model + A1matrix[0][0] - 4 * lambda_w;
Anew[0][3] = lambda_w*lambda_w*lambda_w*lambda_w / (A2matrix[0]
[0] * beta_model) + 1;
Anew[0][4] = A1matrix[0][0] - lambda_inner;
```

```
// Augmented version (currently active in PT3DView.cpp line
3330-3331)
tune_L1 = -(-beta_model + (1 + beta_model)*(1 + beta_model)
- 4 * (1 + beta_model)*lambda_w + 6 *
lambda_w*lambda_w
- lambda_w*lambda_w*lambda_w*lambda_w / beta_model);

tune_L2 = -( 1 + beta_model - 4 * lambda_w
- (1 +
beta_model)*lambda_w*lambda_w*lambda_w*lambda_w
/ (beta_model*beta_model)
+ 4 * lambda_w*lambda_w*lambda_w / beta_model);

tune_L3 = pdlg6->ThreeD_sine_theta;
```

Augmented Version (活躍版本):

$$\text{tune}_{L1} = -\left(-\beta + (1 + \beta)^2 - 4(1 + \beta)\lambda + 6\lambda^2 - \frac{\lambda^4}{\beta}\right)$$

$$\text{tune}_{L2} = -\left(1 + \beta - 4\lambda - (1 + \beta)\frac{\lambda^4}{\beta^2} + 4\frac{\lambda^3}{\beta}\right)$$

$$\text{tune}_{L3} = \text{ThreeD_sine_theta}$$

其中： $\beta = \text{beta_model}$, $\lambda = \text{lambda_w}$

inv_B

```
inv_B = [  
    [1254.4, 365.9, 362.7, 509.3, 510.7, 478.1],  
    [ 354.7, 1162.4, 387.9, 669.9, 692.5, 448.6],  
    [ 427.5, 453.9, 1258.9, 439.3, 488.0, 520.8],  
    [ 442.5, 654.6, 353.8, 1460.7, 800.5, 448.8],  
    [ 414.2, 572.5, 337.4, 687.5, 1274.4, 374.7],  
    [ 412.2, 400.9, 368.5, 474.4, 407.4, 1371.6]  
    * tune_L3;
```

Almatrix_s_lambda

```
Almatrix_s_lambda = [  
    [1254.4, 365.9, 362.7, 509.3, 510.7, 478.1],  
    [ 354.7, 1162.4, 387.9, 669.9, 692.5, 448.6],  
    [ 427.5, 453.9, 1258.9, 439.3, 488.0, 520.8],  
    [ 442.5, 654.6, 353.8, 1460.7, 800.5, 448.8],  
    [ 414.2, 572.5, 337.4, 687.5, 1274.4, 374.7],  
    [ 412.2, 400.9, 368.5, 474.4, 407.4, 1371.6]  
] * tune_L1 * tune_L3;
```

L2_esti

```
L2_esti = [  
    [1254.4, 365.9, 362.7, 509.3, 510.7, 478.1],  
    [ 354.7, 1162.4, 387.9, 669.9, 692.5, 448.6],  
    [ 427.5, 453.9, 1258.9, 439.3, 488.0, 520.8],  
    [ 442.5, 654.6, 353.8, 1460.7, 800.5, 448.8],  
    [ 414.2, 572.5, 337.4, 687.5, 1274.4, 374.7],  
    [ 412.2, 400.9, 368.5, 474.4, 407.4, 1371.6]  
] * tune_L2 * tune_L3;
```

$$\mathbf{u}[k] = \mathbf{B}_{\text{tune}}^{-1} \{ \mathbf{v}_{\text{ff}}[k] + \delta \mathbf{v}_{\text{fb}}[k] \} - \hat{\mathbf{w}}[k]$$

$$\mathbf{v}_{\text{ff}}[k-1] = \mathbf{v}_d[k] - a_1 \mathbf{v}_d[k-1] - a_2 \mathbf{v}_d[k-2]$$

$$\delta \mathbf{v}_{\text{fb}}[k] = (a_1 - \lambda_c) \widehat{\delta \mathbf{v}}[k] + a_2 \widehat{\delta \mathbf{v}}[k-1]$$

$$\hat{\mathbf{s}}_1[k] = \lambda_c \hat{\mathbf{s}}_1[k-1] + \mathbf{L}_1 \{ \delta \mathbf{v}[k-1] - \hat{\mathbf{s}}_1[k-1] \}$$

$$\hat{\mathbf{s}}_2[k] = \hat{\mathbf{s}}_1[k-1] + \mathbf{L}_2 \{ \delta \mathbf{v}[k-1] - \hat{\mathbf{s}}_1[k-1] \}$$

$$\hat{\mathbf{w}}[k] = \hat{\mathbf{w}}[k-1] + \delta \hat{\mathbf{w}}[k-1] + \mathbf{L}_3 \{ \delta \mathbf{v}[k-1] - \hat{\mathbf{s}}_1[k-1] \}$$

$$\delta \hat{\mathbf{w}}[k] = \delta \hat{\mathbf{w}}[k-1] + \mathbf{L}_4 \{ \delta \mathbf{v}[k-1] - \hat{\mathbf{s}}_1[k-1] \}$$

$$L_1 = 1 + \beta + a_1 + 4\lambda_e$$

$$L_2 = 1 + \frac{\lambda_e^4}{a_2 \beta}$$

$$L_3 = - \left(-\beta + (1 + \beta)^2 - 4(1 + \beta)\lambda_e + 6\lambda_e^2 - \frac{\lambda_e^4}{\beta} \right) \mathbf{B}_{\text{tune}}^{-1}$$

$$L_4 = - \left(1 + \beta - 4\lambda_e - (1 + \beta) \frac{\lambda_e^4}{\beta^2} + 4 \frac{\lambda_e^3}{\beta} \right) \mathbf{B}_{\text{tune}}^{-1}$$

$$* \beta = \lambda_e^2$$

HallSensor_VoltCtrl Module Implementation

(inew_control_en Mode)

Part 1: Variable Calculation Timeline [Channel 0]

Case 0: Start

```
MultA1 <= a1; MultB1 <= HsVd1_VECT6[0]  
MultA2 <= a2; MultB2 <= HsVd2_VECT6[0]  
MultA3 <= S1_d1_esti[0]; MultB3 <= lambda_c  
SubA1 <= HsVd2_VECT6[0]; SubB1 <= HsVm1_VECT6[0]
```

Case 6: Result

```
a1_m_Vd_d1[0] <= MultRst1  
a2_m_Vd_d2[0] <= MultRst2  
lamc_M_S1_d1_esti[0] <= MultRst3
```

Case 8: Result

```
S1_d1[0] <= SubRst1
```

Case 9: Start

```
SubA1 <= S1_d1[0]; SubB1 <= S1_d1_esti[0]
```

Case 17: Result

```
Output_diff[0] <= SubRst1
```

Case 18: Start

```
MultA3 <= L1; MultB3 <= Output_diff[0]
```

Case 21: Start

```
MultA1 <= iHsVd_VECT6[0]; MultB1 <= HsVd_Coeff  
MultA2 <= HsVd1_VECT6[0]; MultB2 <= HsVd_Coeff  
SubA1 <= iHsVd_VECT6[0]; SubB1 <= a1_m_Vd_d1[0]
```

Case 22: Start

```
MultA1 <= HsVd1_LPF_VECT6[0]; MultB1 <= HsVd_LPF_Coeff1
```

Case 24: Result

$S1_esti_correct[0] \leftarrow MultRst3$

$SubA1 \leftarrow Vd_s_a1Vd1[0]; SubB1 \leftarrow a2_m_Vd_d2[0]$

Case 25: Start

$AddA2 \leftarrow lamc_M_S1_d1_esti[0]; AddB2 \leftarrow S1_esti_correct[0]$

Case 27: Result

$HsVd_Coeff_M_HsVd_VECT6[0] \leftarrow MultRst1$

$HsVd_Coeff_M_HsVd1_VECT6[0] \leftarrow MultRst2$

$AddA1 \leftarrow MultRst1; AddB1 \leftarrow MultRst2$

$MultA3 \leftarrow L2; MultB3 \leftarrow Output_diff[0]$

Case 28: Result

$HsVd_LPF_Coeff1_M_HsVd1_LPF_VECT6[0] \leftarrow MultRst1$

Case 32: Result

$Vd_s_a1Vd1_s_a2Vd2[0] \leftarrow SubRst1$

$S1_esti[0] \leftarrow AddRst2$

Case 33: Start

$AddA2 \leftarrow S1_d1_esti[0]; AddB2 \leftarrow S2_esti_correct[0]$

$MultA2 \leftarrow a1_s_lamc; MultB2 \leftarrow S1_esti[0]$

Case 35: Result

$HsVd_ForLPF_Sum_VECT6[0] \leftarrow AddRst1$

$AddA1 \leftarrow AddRst1; AddB1 \leftarrow HsVd_LPF_Coeff1_M_HsVd1_LPF_VECT6[0]$

Case 41: Result

$a1lamc_m_S1_esti[0] \leftarrow MultRst2$

$AddA2 \leftarrow Vd_s_a1Vd1_s_a2Vd2[0]; AddB2 \leftarrow a1lamc_m_S1_esti[0]$

$S2_esti[0] \leftarrow AddRst2$

Case 43: Result

$HsVd_LPF_VECT6[0] \leftarrow AddRst1$

$MultA2 \leftarrow a2; MultB2 \leftarrow S2_esti[0]$

Case 49: Result

Vd_s_a1Vd_s_a2Vd2_p_allamc_m_S1_esti[0] <= AddRst2
a2_m_S2_esti[0] <= MultRst2

Case 50: Start

AddA2 <= Vd_s_a1Vd_s_a2Vd2_p_allamc_m_S1_esti[0]; AddB2 <= a2_m_S2_esti[0]

Case 58: Result

u_before_Btuneinv[0] <= AddRst2

Case 64: Start

mVolt_6x1_3 <= u_before_Btuneinv
mTransf_MAT6x6_3 <= i_inv_B_tune

Case 68: Start

MultA4 <= L3; MultB4 <= Output_diff[0]

Case 71: Start

westi_before_correct[0] <= AddRst2

Case 74: Result

westi_correct[0] <= MultRst4

Case 80: Start

MultA4 <= L4; MultB4 <= Output_diff[0]

Case 86: Result

delta_westi_correct[0] <= MultRst4

Case 93: Start

AddA2 <= westi_before_correct[0]; AddB2 <= westi_correct[0]

Case 101: Result

westi[0] <= AddRst2

Case 107: Start

mVolt_6x1_4 <= westi
mTransf_MAT6x6_4 <= i_inv_B

Case 110: Start

AddA2 <= delta_westi_d1[0]; AddB2 <= delta_westi_correct[0]

Case 116: Result

delta_westi[0] <= AddRst2

Case 133: Result

u_before_w <= Rst_6x6_6x1_3

Case 176: Result

westi_after_Binv <= Rst_6x6_6x1_4

Case 177: Start

SubA5 <= u_before_w[0]; SubB5 <= westi_after_Binv[0]

Case 185: Result

u_total_new[0] <= SubRst5

Case 191: Result

ou_new[0] <= u_total_new[0]

Case 207: Start

MultA3 <= ou_new[0]; MultB3 <= iDA_Scale1V_fp

Case 213: Result

u_new_DAscale[0] <= MultRst3

Case 217: Start

FP2INT_data <= u_new_DAscale[0]

Case 224: Result

u_new_DAscale_int[0] <= FP2INT_Rst

ou_new_DAscale_int_offset[0] <= FP2INT_Rst + 32768

Part 2: Mathematical Equations

$S1_d1 = HsVd2_VECT6 - HsVm1_VECT6$

$Output_diff = S1_d1 - S1_d1_esti$

$a1_m_Vd_d1 = a1 * HsVd1_VECT6$

```

a2_m_Vd_d2 = a2 * HsVd2_VECT6
lamc_M_S1_d1_esti = lambda_c * S1_d1_esti

Vd_s_a1Vd1 = iHsVd_VECT6 - a1_m_Vd_d1
Vd_s_a1Vd1_s_a2Vd2 = Vd_s_a1Vd1 - a2_m_Vd_d2

S1_esti_correct = L1 * Output_diff
S2_esti_correct = L2 * Output_diff
S1_esti = lamc_M_S1_d1_esti + S1_esti_correct
S2_esti = S1_d1_esti + S2_esti_correct

allamc_m_S1_esti = a1_s_lamc * S1_esti
a2_m_S2_esti = a2 * S2_esti

Vd_s_a1Vd_s_a2Vd2_p_allamc_m_S1_esti = Vd_s_a1Vd1_s_a2Vd2 +
allamc_m_S1_esti
u_before_Btuneinv = Vd_s_a1Vd_s_a2Vd2_p_allamc_m_S1_esti +
a2_m_S2_esti

westi_correct = L3 * Output_diff
delta_westi_correct = L4 * Output_diff
westi_before_correct = westi_d1 + delta_westi_d1
westi = westi_before_correct + westi_correct
delta_westi = delta_westi_d1 + delta_westi_correct

u_before_w = i_inv_B_tune * u_before_Btuneinv
westi_after_Binv = i_inv_B * westi
u_total_new = u_before_w - westi_after_Binv
ou_new = u_total_new

u_new_DAscale = ou_new * iDA_Scale1V_fp
ou_new_DAscale_int_offset = FP2INT(u_new_DAscale) + 32768

```


Part 3: Low-Pass Filter Variables (For Completeness)

HsVd_Coeff = iHsCtrl_SplIntv / (2*iHsVd_LPF_tau + iHsCtrl_SplIntv)

**HsVd_LPF_Coeff1 = (2*iHsVd_LPF_tau - iHsCtrl_SplIntv) /
(2*iHsVd_LPF_tau + iHsCtrl_SplIntv)**

HsVd_Coeff_M_HsVd_VECT6 = iHsVd_VECT6 * HsVd_Coeff

HsVd_Coeff_M_HsVd1_VECT6 = HsVd1_VECT6 * HsVd_Coeff

HsVd_LPF_Coeff1_M_HsVd1_LPF_VECT6 = HsVd1_LPF_VECT6 *

HsVd_LPF_Coeff1

HsVd_ForLPF_Sum_VECT6 = HsVd_Coeff_M_HsVd_VECT6 +

HsVd_Coeff_M_HsVd1_VECT6

HsVd_LPF_VECT6 = HsVd_ForLPF_Sum_VECT6 +

HsVd_LPF_Coeff1_M_HsVd1_LPF_VECT6