PID Control Implementation Record

1. Variable Operations Record

Low-Pass Filter Coefficient Calculation

case 5′d21:

```
TwoTau = 2 * iHsVd_LPF_tau
TwoTau_A_T = TwoTau + iHsCtrl_SplIntv
TwoTau_S_T = TwoTau - iHsCtrl_SplIntv
HsVd_Coeff = iHsCtrl_SplIntv / TwoTau_A_T
```

case 5′d29:

```
HsVd_LPF_Coeff1 = TwoTau_S_T / TwoTau_A_T
```

Low-Pass Filter Operations

case 9′d21:

```
HsVd_Coeff_M_HsVd_VECT6_0 = iHsVd_VECT6[0] * HsVd_Coeff
HsVd_Coeff_M_HsVd1_VECT6_0 = HsVd1_VECT6[0] * HsVd_Coeff
```

case 9′d22:

```
HsVd_LPF_Coeff1_M_HsVd1_LPF_VECT6_0 = HsVd1_LPF_VECT6[0] * HsVd_LPF_Coeff1
```

case 9′d35:

```
HsVd_ForLPF_Sum_VECT6_0 = HsVd_Coeff_M_HsVd_VECT6_0 + HsVd_Coeff_M_HsVd1_VECT6_0
HsVd_LPF_VECT6_0 = HsVd_ForLPF_Sum_VECT6_0 + HsVd_LPF_Coeff1_M_HsVd1_LPF_VECT6_0
```

PID Coefficient Calculation

case 5′d0:

```
HsIgain_M_SplIntvO2_VECT6[0] = iHsIgain_VECT6[0] * iHsCtrl_SplIntvO2
HsDgain_D_SplIntvO2_VECT6[0] = iHsDgain_VECT6[0] / iHsCtrl_SplIntvO2
HsIgain_M_SplIntv_VECT6[0] = iHsIgain_VECT6[0] * iHsCtrl_SplIntv
```

case 5′d7:

```
Intm_Coeff_VECT6[0] = HsDgain_D_SplIntvO2_VECT6[0] + HsIgain_M_SplIntvO2_VECT6[0]
```

case 5′d13:

```
FourHsDgain_D_SplIntv_VECT6[0] = 2 * HsDgain_D_SplIntvO2_VECT6[0]
```

case 5′d15:

```
oHsCoeff_VECT6[0] = Intm_Coeff_VECT6[0] + iHsPgain_VECT6[0]
oHsCoeff1_VECT6[0] = Intm_Coeff_VECT6[0] - iHsPgain_VECT6[0]
```

case 5′d27:

```
oHsCoeff2_VECT6[0] = HsIgain_M_SplIntv_VECT6[0] - FourHsDgain_D_SplIntv_VECT6[0]
```

Error Calculation

case 9′d56:

```
oHsVerr_VECT6[0] = HsVd_LPF_VECT6[0] - iHsVm_VECT6[0]
```

case 9′d63:

```
oHsVctrlFF_VECT6[0] = HsVd_LPF_VECT6[0] / iHsFFgain_VECT6[0]
```

PID Control Calculation

case 9′d64:

```
HsVerr_M_HsCoeff_VECT6[0] = oHsVerr_VECT6[0] * oHsCoeff_VECT6[0]
```

case 9′d70:

```
HsVerrHsCoeff_A_HsVctrl2_VECT6[0] = HsVerr_M_HsCoeff_VECT6[0] +
HsVctrlCompl_2_VECT6[0]
```

case 9′d76:

```
HsVerr1_M_HsCoeff1_VECT6[0] = HsVerr1_VECT6[0] * oHsCoeff1_VECT6[0]
```

case 9′d82:

```
HsVerr2_M_HsCoeff2_VECT6[0] = HsVerr2_VECT6[0] * oHsCoeff2_VECT6[0]
```

case 9′d90:

```
HsVerr1HsCoeff1_A_HsVerr2HsCoeff2_VECT6[0] = HsVerr2_M_HsCoeff2_VECT6[0] +
HsVerr1_M_HsCoeff1_VECT6[0]
```

case 9′d98:

```
oHsVctrlCompl_VECT6[0] = HsVerr1HsCoeff1_A_HsVerr2HsCoeff2_VECT6[0] +
HsVerrHsCoeff_A_HsVctrl2_VECT6[0]
```

```
oHsVctrlTot_VECT6[0] = oHsVctrlCompl_VECT6[0] + oHsVctrlFF_VECT6[0]
```

Low-Pass Filter Difference Equation

$$Vd_{LPF}[k] = \frac{T}{2\tau+T} \cdot (Vd[k] + Vd[k-1]) + \frac{2\tau-T}{2\tau+T} \cdot Vd_{LPF}[k-1]$$

Where:

- $T = \frac{1}{100000}$ (Sampling period = 10 μs)
- $\tau = \frac{1}{10000}$ (Low-pass filter time constant = 100 μs)

PID Control Difference Equation

$$e[k] = Vd_{LPF}[k] - Vm[k]$$

$$\begin{aligned} u_{PID}[k] = &\left(K_p + K_i \cdot \frac{T}{2} + \frac{K_d}{\frac{T}{2}}\right) \cdot e[k] & + \left(K_i \cdot \frac{T}{2} + \frac{K_d}{\frac{T}{2}} - K_p\right) \cdot e[k-1] \\ &+ \left(K_i \cdot T - 4 \cdot \frac{K_d}{T}\right) \cdot e[k-2] & + u[k-2] \end{aligned}$$

Where:

- $K_p = 8$
- $K_i = 20000$
- $K_d = 0$

Total Control Output

$$u_{total}[k] = u_{PID}[k] + u_{FF}[k]$$

$$u_{FF}[k] = Vd_{LPF}[k] / F_{gain}$$

Feedforward Gains:

- Channel 0: $F_0 = 0.5716$
- Channel 1: $F_1 = 0.5832$
- Channel 2: $F_2 = 0.5945$
- Channel 3: $F_3 = 0.5389$
- Channel 4: $F_4 = 0.6081$
- Channel 5: $F_5 = 0.5622$

DAC Output

$$\boldsymbol{V}_{out} = \boldsymbol{M}_{DC} \cdot \boldsymbol{u}_{total}$$

Timing Marks

case 5′d0 → case 5′d10: PID coefficient calculation completed

case 5′d21 → case 5′d29: Low-Pass Filter coefficient calculation completed

case 9′d56 → case 9′d98: PID control operation completed, output oHsVctrlTot_VECT6[0]

Extended PID Control Analysis

3. Complete PID Implementation Formula

Tustin Transform Coefficients

The PID controller uses Tustin (bilinear) transform to discretize the continuous PID controller:

Coefficient Calculations:

```
Intm_Coeff = Kd/(T/2) + Ki*(T/2)
oHsCoeff = Intm_Coeff + Kp = Kp + Ki*(T/2) + Kd/(T/2)
oHsCoeff1 = Intm_Coeff - Kp = Ki*(T/2) + Kd/(T/2) - Kp
oHsCoeff2 = Ki*T - 4*Kd/T
```

Complete PID Recursive Formula

The discrete PID controller implementation:

```
u[k] = oHsCoeff * e[k] + oHsCoeff1 * e[k-1] + oHsCoeff2 * e[k-2] + u[k-2]
```

Where:

- `e[k] = Vd_LPF[k] - Vm[k]` (Error signal)
- `u[k]` = Control output
- `e[k-1], e[k-2]` = Previous error samples
- `u[k-2]` = Control output two samples ago

Low-Pass Filter Implementation

The desired voltage `Vd` passes through a low-pass filter before PID control:

```
Vd_LPF[k] = HsVd_Coeff * (Vd[k] + Vd[k-1]) + HsVd_LPF_Coeff1 * Vd_LPF[k-1]
```

Filter coefficients:

- `HsVd_Coeff = T/(2*tau + T)`
- `HsVd_LPF_Coeff1 = (2*tau - T)/(2*tau + T)`

Feedforward Control

The total control effort includes feedforward compensation:

```
u_total[k] = u_PID[k] + u_FF[k]
```

Where feedforward term: `u_FF[k] = Vd_LPF[k] / FFgain`

4. Implementation Details

Sampling Rate

- Default sampling frequency: 100 kHz (T = 10 μs)
- Optional 200 kHz mode available via `mHsCtrl_Rate` flag

History Management

The controller maintains history for recursive computation:

- `HsVerr1_VECT6[i]` = e[k-1]
- `HsVerr2_VECT6[i]` = e[k-2]
- `HsVctrlCompl_1_VECT6[i]` = u[k-1]
- `HsVctrlCompl_2_VECT6[i]` = u[k-2]

Parameter Update Handling

When PID gains or sampling interval changes:

1. All history buffers reset to zero
2. Coefficients recalculated (case 5′d0 to 5′d29)
3. `HsCoeff_InHsCtrl_bol` flag set to trigger update

Output Scaling

Final control output scaled for DAC:

1. Multiply by `iDA_Scale1V_fp` (floating-point scaling)
2. Convert to 16-bit integer
3. Add offset 32768 for unsigned DAC format

5. State Machine Sequence

Coefficient Update Phase (5-bit counter)

- **States 0-5:** Calculate `HsIgain_M_SplIntv02`, `HsDgain_D_SplIntv02`
- **States 6-12:** Calculate intermediate coefficients
- **States 13-20:** Calculate `FourHsDgain_D_SplIntv`, `Intm_Coeff`
- **States 21-29:** Final coefficient assembly and L P F coefficients

Control Calculation Phase (9-bit counter)

- **States 0-20:** State estimation and prediction
- **States 21-55:** Low-pass filter calculation
- **States 56-69:** Error calculation and feedforward
- **States 70-98:** PID control law computation
- **States 99-111:** Total control output assembly
- **States 112-230:** Output scaling and format conversion

PID Control Mathematical Model with System Parameters

6. Complete Mathematical Formulation

System Parameters (from USB_Parameter_Transmission)

Sampling Parameters:

- Sampling frequency: $f_s = 100$ kHz (optional 200 kHz via `mHsCtrl_Rate`)
- Sampling period: $T = \frac{1}{f_s} = 10$ μs
- Half sampling period: $\frac{T}{2} = 5$ μs

Low-Pass Filter:

- Time constant: $\tau = \frac{1}{10000} = 0.1$ ms
- Filter coefficient 1: $\alpha_1 = \frac{T}{2\tau+T} = \frac{0.01}{0.2+0.01} = 0.0476$
- Filter coefficient 2: $\alpha_2 = \frac{2\tau-T}{2\tau+T} = \frac{0.2-0.01}{0.2+0.01} = 0.9048$

Default PID Gains (6 channels):

- Proportional gain: $K_p = 8.0$
- Integral gain: $K_i = 20000.0$
- Derivative gain: $K_d = 0.0$

Feedforward Gains (6 channels):

- Channel 0: $\mathbb{F}_0 = 0.5716$
- Channel 1: $\mathbb{F}_1 = 0.5832$
- Channel 2: $\mathbb{F}_2 = 0.5945$
- Channel 3: $\mathbb{F}_3 = 0.5389$
- Channel 4: $\mathbb{F}_4 = 0.6081$
- Channel 5: $\mathbb{F}_5 = 0.5622$

Complete Control System Equations

1. Low-Pass Filter (Input Smoothing)

$V_{d_{LPF}}[k] = \alpha_1\left(V_{d[k]} + V_{d[k-1]}\right) + \alpha_2 V_d^{LPF}[k-1]$

Substituting parameters: $V_{d_{LPF}}[k] = 0.0476\left(V_{d[k]} + V_{d[k-1]}\right) + 0.9048 V_d^{LPF}[k-1]$

2. Error Signal

$e[k] = V_d^{LPF}[k] - V_{m[k]}$

Where:

- $V_{d_{LPF}}[k]$: Filtered desired voltage
- $V_{m[k]}$: Measured voltage from Hall sensor

3. PID Control Coefficients

With $T = 10 \times 10^{-6}$ s, $K_p = 8$, $K_i = 20000$, $K_d = 0$:

$C_0 = K_p + K_i \frac{T}{2} + \frac{K_d}{\frac{T}{2}} = 8 + 20000 \times 5 \times 10^{-6} + 0 = 8.1$

$C_1 = K_i \frac{T}{2} + \frac{K_d}{\frac{T}{2}} - K_p = 20000 \times 5 \times 10^{-6} + 0 - 8 = -7.9$

$C_2 = K_i T - 4\frac{K_d}{T} = 20000 \times 10 \times 10^{-6} - 0 = 0.2$

4. Discrete PID Controller

$u_{PID}[k] = C_0 e[k] + C_1 e[k-1] + C_2 e[k-2] + u[k-2]$

Substituting coefficients: $u_{PID}[k] = 8.1e[k] - 7.9e[k-1] + 0.2e[k-2] + u[k-2]$

5. Feedforward Control

$u_{FF}[k] = V_d^{LPF}\frac{[k]}{\mathbb{F}_i}$

Where $\mathbb{F}_i$ is the feedforward gain for channel $i \in \{0,1,2,3,4,5\}$

6. Total Control Output

$u_{total}[k] = u^{PID}[k] + u^{FF}[k]$

7. Multi-Channel Matrix Operation

DC Transformation Matrix (Normalized)

The system uses a 6×6 DC matrix for multi-channel control coordination:

$$\boldsymbol{M}_{DC} = \begin{pmatrix} 0.482 & 0.040 & 0.069 & 0.106 & 0.097 & 0.064 \\ 0.057 & 0.314 & 0.129 & 0.072 & 0.057 & 0.134 \\ 0.068 & 0.092 & 0.460 & 0.047 & 0.083 & 0.071 \\ 0.127 & 0.061 & 0.057 & 0.380 & 0.060 & 0.123 \\ 0.136 & 0.057 & 0.120 & 0.071 & 0.311 & 0.056 \\ 0.063 & 0.096 & 0.071 & 0.103 & 0.038 & 0.465 \end{pmatrix}$$

Channel Coupling

For 6-channel simultaneous control: $\boldsymbol{V}_{out} = \boldsymbol{M}_{DC} \cdot \boldsymbol{U}_{total}$

Where:

- $\boldsymbol{U}_{total} = \left[u_{0_{total}}, u_{1_{total}}, ..., u_{5_{total}}\right]^T$: Control outputs
- $\boldsymbol{V}_{out} = \left[V_{0_{out}}, V_{1_{out}}, ..., V_{5_{out}}\right]^T$: Applied voltages

8. Implementation Verification Checklist

Coefficient Calculations

☐ Low-pass filter coefficient $\alpha_1 = \frac{T}{2\tau+T}$ ✓ ☐ Low-pass filter coefficient $\alpha_2 = \frac{2\tau-T}{2\tau+T}$ ✓ ☐ PID coefficient $C_0 = K_p + K_i\frac{T}{2} + \frac{K_d}{\frac{T}{2}}$ ✓ ☐ PID coefficient $C_1 = K_i\frac{T}{2} + \frac{K_d}{\frac{T}{2}} - K_p$ ✓ ☐ PID coefficient $C_2 = K_iT - 4\frac{K_d}{T}$ ✓

Signal Flow

1. Input $V_{d[k]}$ ⟶ Low-pass filter ⟶ $V_{d_{LPF}}[k]$
2. Error calculation: $e[k] = V_d^{LPF}[k] - V_{m[k]}$
3. PID control: $u_{PID}[k]$ using $e[k], e[k-1], e[k-2], u[k-2]$
4. Feedforward: $u_{FF}[k] = V_d^{LPF}\frac{[k]}{\mathbb{F}_i}$
5. Total output: $u_{total}[k] = u^{PID}[k] + u^{\mathbb{F}}[k]$
6. Matrix transformation: $\boldsymbol{V}_{out} = \boldsymbol{M}_{DC} \cdot \boldsymbol{U}_{total}$

Parameter Ranges

- Sampling rate: 100-200 kHz
- P-gain: 0-100 (typical: 8.0)
- I-gain: 0-100000 (typical: 20000.0)
- D-gain: 0-10 (typical: 0.0)
- L P F time constant: 0.1 ms (fixed)
- Feedforward gains: 0.5-0.7 (channel-specific)