

# PID Feedback Controller & IMC-PID Tuning Rules

## FOPDT Process Model

$$G_p(s) = \frac{K_p \cdot e^{-\theta s}}{T_p s + 1} ; K_p : \text{Process Gain} ;$$

$\theta : \text{Dead Time } T_p : \text{Time Constant}$

## PID (Real) Controller Algorithm

$$G_c(s) = K_c \left( 1 + \frac{1}{T_i s} + T_d s \frac{1}{T_f s + 1} \right)$$

## IMC – PID Tuning Rules

$\lambda : \text{Desired Closed Loop Time Constant}$

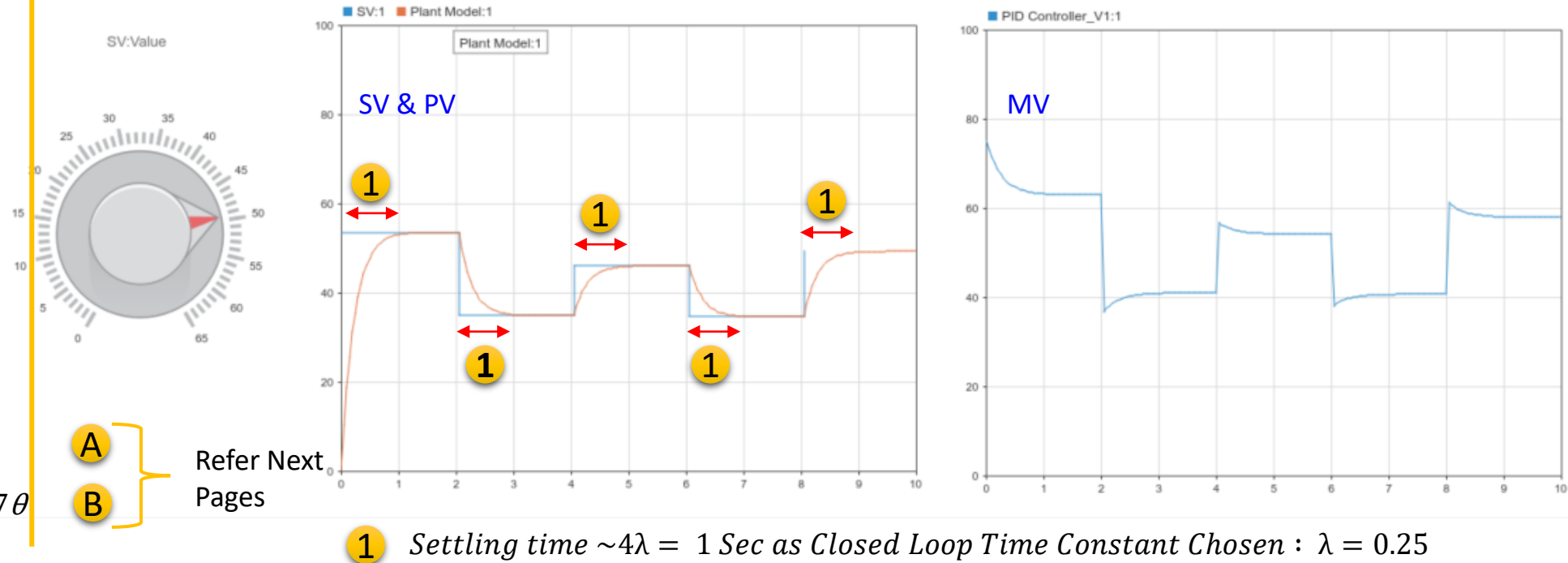
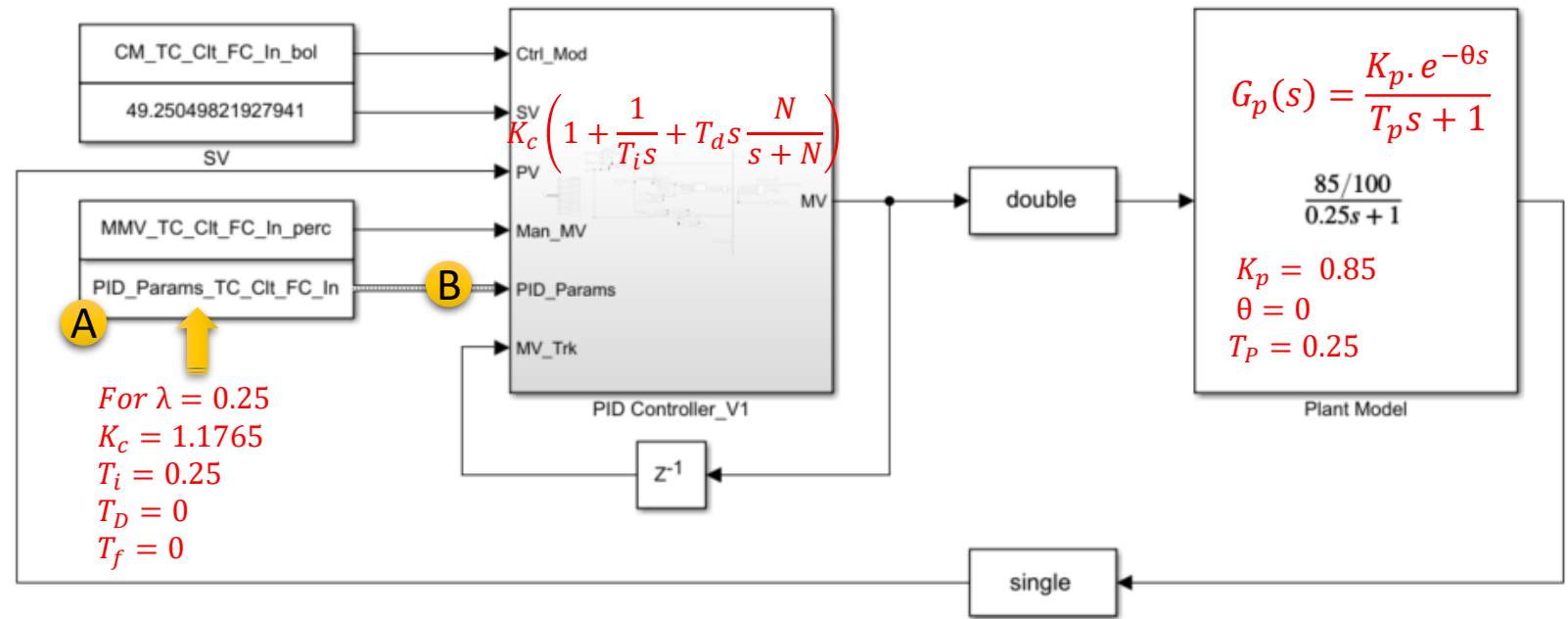
$$K_c = \frac{T_p + 0.5 * \theta}{K_p(\theta + \lambda)}$$

$$T_i = T_p + 0.5 * \theta$$

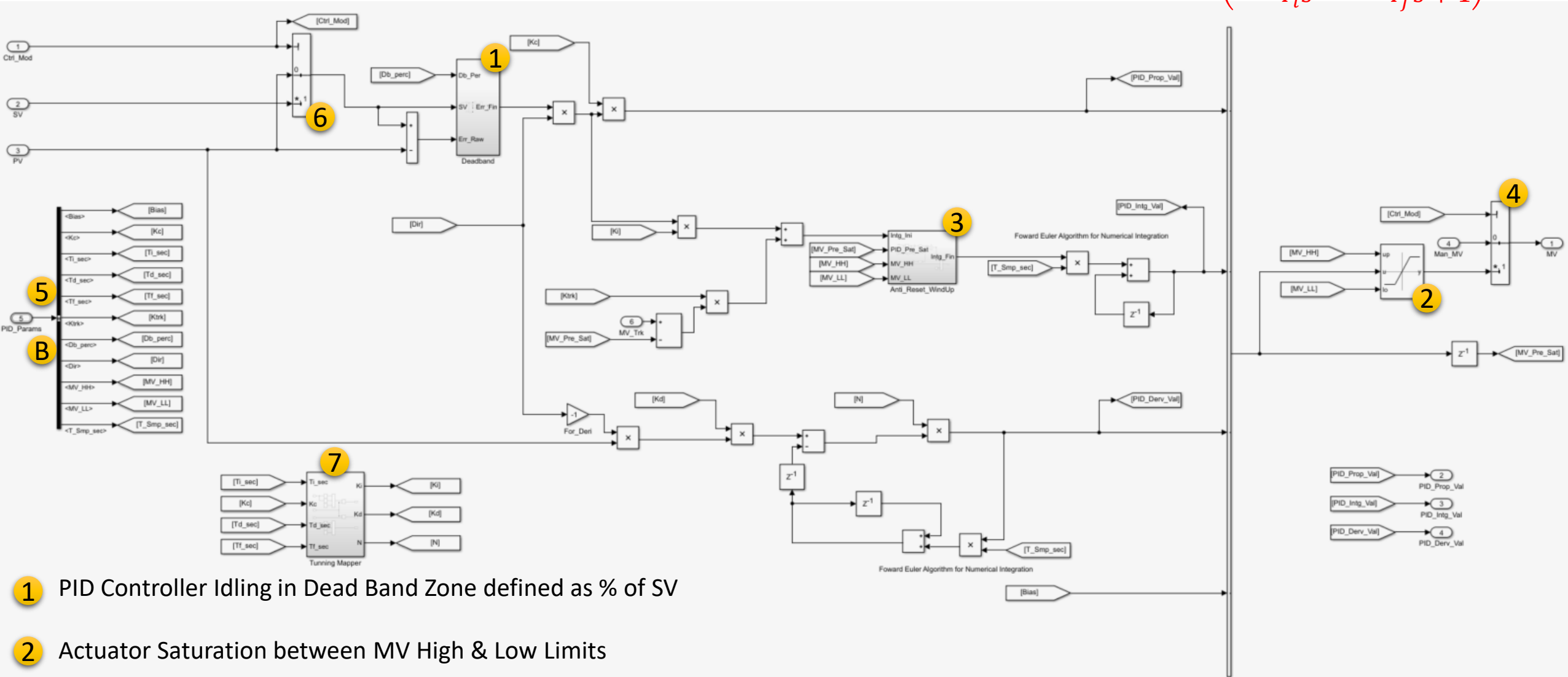
$$T_d = \frac{T_p \cdot \theta}{2 \cdot T_p + \theta}$$

$$T_f = \frac{\theta \cdot \lambda}{2(\theta + \lambda)}$$

Recommended :  $\lambda > 0.2 T_p$  &  $\lambda > 1.7 \theta$



# Real PID Controller Implementation Simulink *(Filtered Derivative on Rate of Change of PV)* $G_c(s) = Bias + K_c \left( 1 + \frac{1}{T_i s} + T_d s \frac{1}{T_f s + 1} \right)$



1 PID Controller Idling in Dead Band Zone defined as % of SV

2 Actuator Saturation between MV High & Low Limits

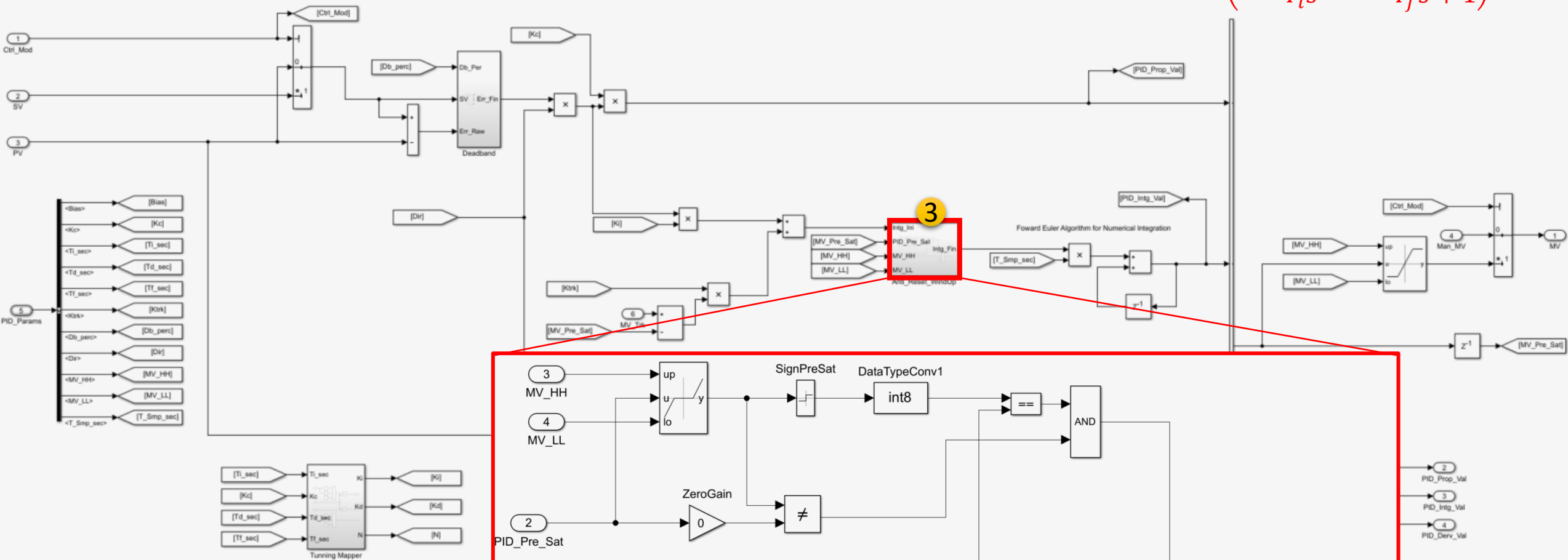
3 Integral Anti Reset Wind Up in the Event of Actuator Saturation

4 PID Controller Output Override in the event Controller in Manual Mode

5 PID Tuning Parameters feed to the Controller via a Bus Signal PID\_Params

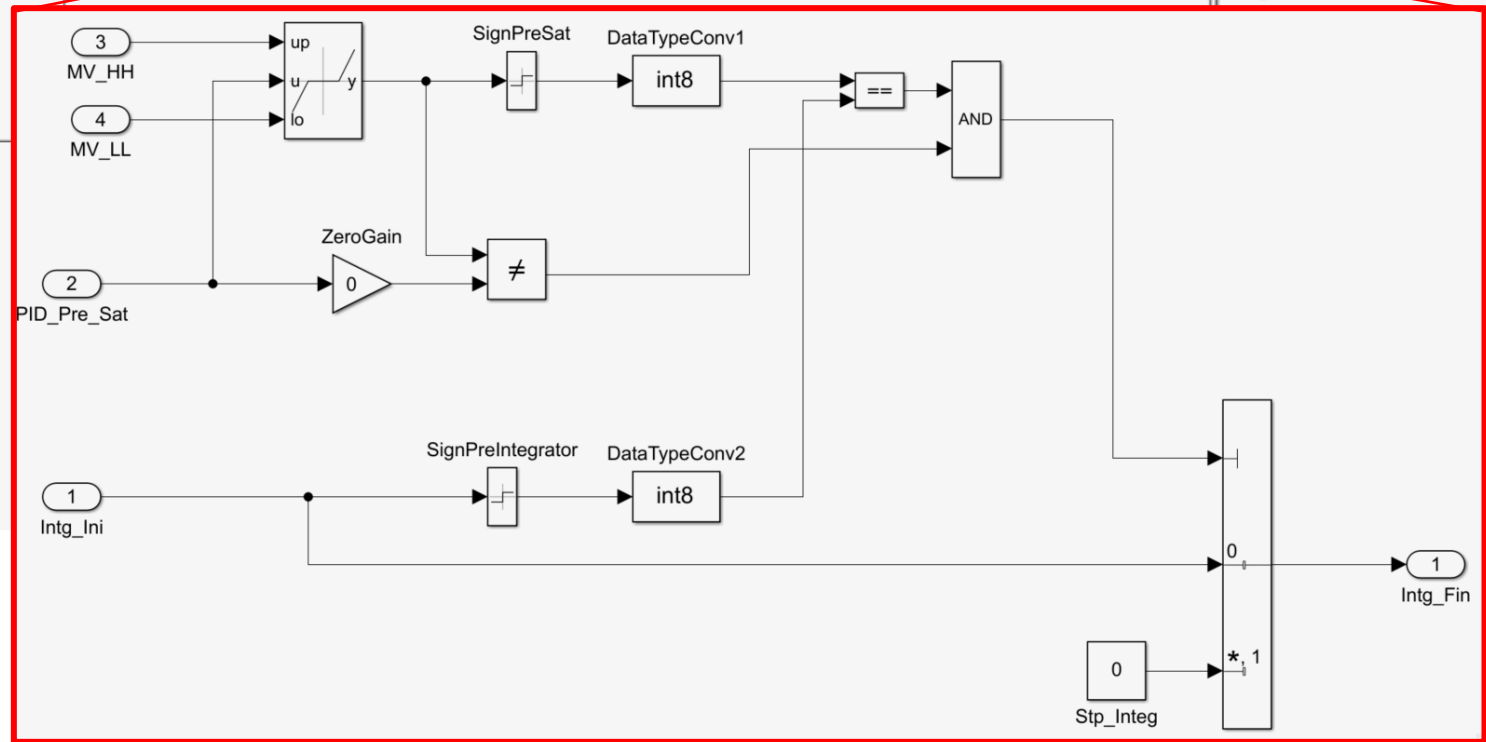
6 In Manual Mode the Error to the PID algorithm is Zero as PV is fed into SV

# Real PID Controller Implementation Simulink *(Filtered Derivative on Rate of Change of PV)* $G_c(s) = Bias + K_c \left( 1 + \frac{1}{T_i s} + T_d s \frac{1}{T_f s + 1} \right)$

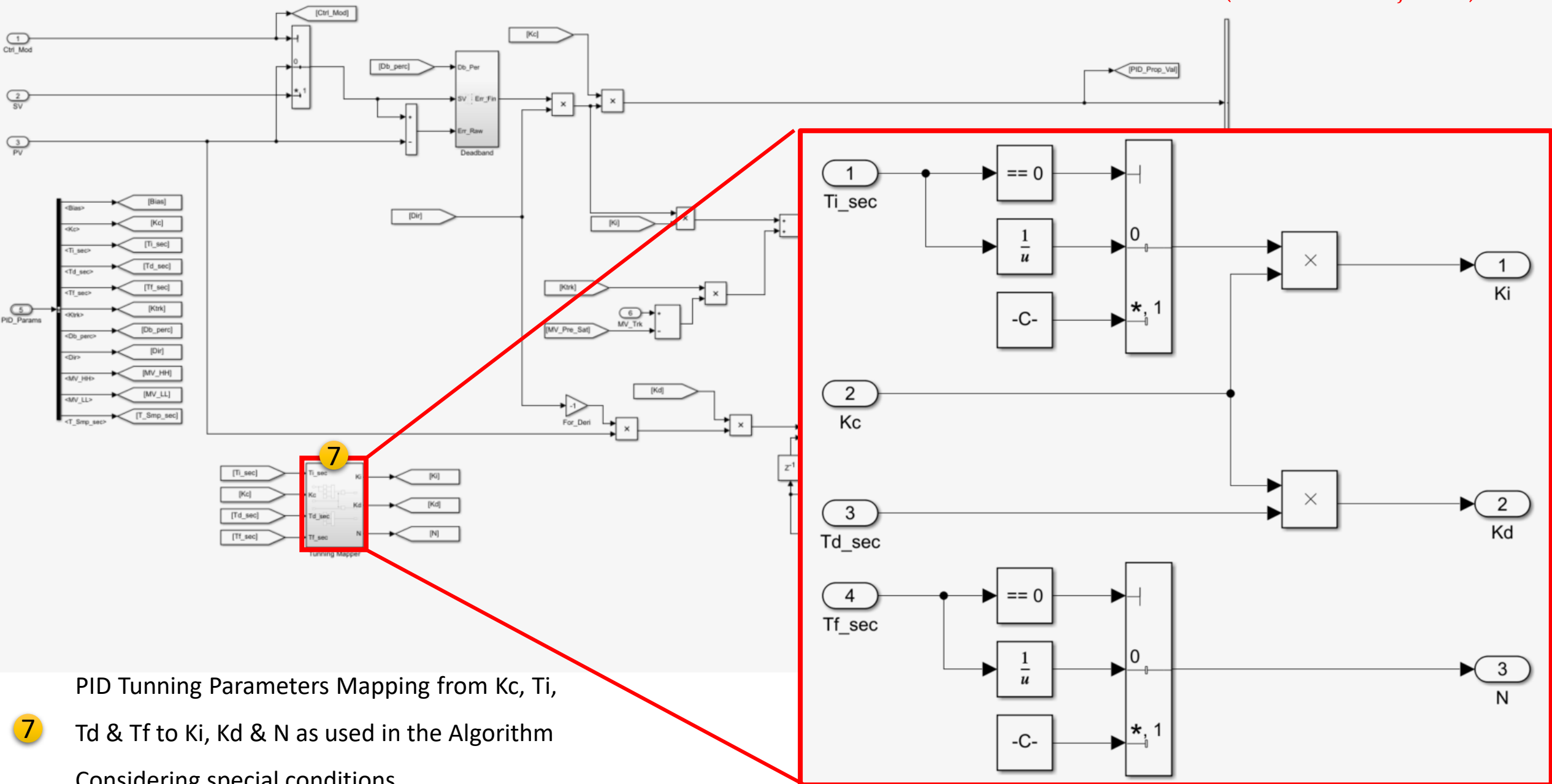


## 3 Integral Anti Reset Wind Up in the Event of Actuator Saturation

In the event of actuator saturation, the code will allow the current Integrator contribution if its sign can help back the pre-saturated final PID within the the Saturation limits



# Real PID Controller Implementation Simulink *(Filtered Derivative on Rate of Change of PV)* $G_c(s) = Bias + K_c \left( 1 + \frac{1}{T_i s} + T_d s \frac{1}{T_f s + 1} \right)$



PID Tuning Parameters Mapping from Kc, Ti, Td & Tf to Ki, Kd & N as used in the Algorithm  
Considering special conditions

**A** % PID Tuning Parameters

```
PID_Params_TC_Clt_FC_In.Bias = single(0);
PID_Params_TC_Clt_FC_In.Kc = single(1.17);
PID_Params_TC_Clt_FC_In.Ti_sec = single(0.25);
PID_Params_TC_Clt_FC_In.Td_sec = single(0);
PID_Params_TC_Clt_FC_In.N = single(0);
PID_Params_TC_Clt_FC_In.Ktrk = single(0);
PID_Params_TC_Clt_FC_In.Db_perc = single(0);
PID_Params_TC_Clt_FC_In.Dir = single(1);
PID_Params_TC_Clt_FC_In.MV_HH = single(100);
PID_Params_TC_Clt_FC_In.MV_LL = single(0);
PID_Params_TC_Clt_FC_In.T_Smp_sec = single(0.05);
```

Bus Editor - Manage Bus Objects in the Base Workspace

File Edit View Options Help

101 010

Base Workspace

- PID\_Params\_Bus\_Obj**
  - Bias
  - Kc
  - Ti\_sec
  - Td\_sec
  - Tf\_sec
  - Ktrk
  - Db\_perc
  - Dir
  - MV\_HH
  - MV\_LL
  - T\_Smp\_sec

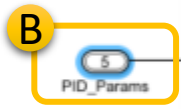
Typical Structure ex

*'PID\_Param\_TC\_Clt\_FC\_In'*

Based Initialization via M File

of the PID Tunning Parameters

loaded on the Workspace.



Typical Bus declaration for

*'PID\_Param\_Bus\_Obj'* as input

declaration PID Controller Block.

To be loaded on the workspace

Block Parameters: PID\_Params

Import

Provide an input port for a subsystem or model. For Triggered Subsystems, 'Latch input by delaying outside signal' produces the value of the subsystem input at the previous time step. For Function-Call Subsystems, turning 'On' the 'Latch input for feedback signals of function-call subsystem outputs' prevents the input value to this subsystem from changing during its execution. The other parameters can be used to explicitly specify the input signal attributes.

Main **Signal Attributes**

☐ Output function call

Minimum: Maximum:

Data type: **Bus: PID\_Params\_Bus\_Obj** >>

☐ Lock output data type setting against changes by the fixed-point tools

Unit (e.g., m, m/s^2, N\*m):

inherit

Port dimensions (-1 for inherited):

-1

Variable-size signal: Inherit

Sample time (-1 for inherited):

-1

Signal type: auto

OK Cancel Help Apply

**Table 7.2 PID Tuning Parameters for First-Order + Time Delay Processes**

Controller	$k_c$	$\tau_I$	$\tau_D$	$\tau_F$	Notes
PID	$\frac{\theta}{k_p(\theta+\lambda)}$	$\tau_p + \frac{\theta}{2}$	$\frac{\tau_p \theta}{2\tau_p + \theta}$	$\frac{\theta \lambda}{2(\theta + \lambda)}$	(1)
PID	$\frac{\theta}{k_p(\lambda + \frac{\theta}{2})}$	$\tau_p + \frac{\theta}{2}$	$\frac{\tau_p \theta}{2\tau_p + \theta}$	—	(2)
PI	$\frac{\tau_p}{k_p \lambda}$	$\tau_p$	—	—	(3)
Improved PI	$\frac{\theta}{k_p \lambda}$	$\tau_p + \frac{\theta}{2}$	—	—	(4)

$$g_p(s) = \frac{k_p e^{-\theta s}}{\tau_p s + 1}$$

$$g_c(s) = k_c \left[ \frac{\tau_I \tau_D s^2 + \tau_I s + 1}{\tau_I s} \right] \left[ \frac{1}{\tau_F s + 1} \right]$$

- (1) With an “all-pass” factorization and semi-proper  $q(s)$ . Recommended  $\lambda > 0.25\theta$ .
- (2) Without an “all-pass” factorization and improper  $q(s)$ . Recommended  $\lambda > 0.8\theta$ .
- (3) With zero-order Padé approximation ( $e^{-\theta s} \approx 1$ ). Recommended  $\lambda > 1.7\theta$ .
- (4) With the approximation  $\frac{k_p e^{-\theta s}}{\tau_p s + 1} \approx \frac{k_p}{(\tau_p + \frac{\theta}{2})s + 1}$ . Recommended  $\lambda > 1.7\theta$ .

In all cases it is recommended that  $\lambda > 0.2 \tau_p$ .

**PID Feedback Controller & IMC-PID Tuning Rules**

Reference : [https://rpi.edu/dept/chem-eng/WWW/faculty/bequette/courses/cpc/IMC\\_PID.pdf](https://rpi.edu/dept/chem-eng/WWW/faculty/bequette/courses/cpc/IMC_PID.pdf)

```
% Process Model Input Parameters

Kp=0.85; % Process Gain in respective Units = [PV MV^-1]
Tp=0.25; % Process Time Constant in Sec
Tdt=0; % Process Dead Time in Sec

% Controller Design Parameter
Lbd = 0.25; % Process Control Closed Loop Desired Time Constant in Sec
% Recommended Lambda should be > 0.2 times Process Time Constant & > 0.8 times Process Dead Time

% Computed Controller Parameters

Kc = (Tp+0.5*Tdt)/(Kp*(Tdt+Lbd)) % Controller Gain in respective Units ie [MV PV^-1]

Kc = 1.1765

Ti = Tp+0.5*Tdt % Controller Integral Time in Secs

Ti = 0.2500

Td = Tp*Tdt/(2*Tp+Tdt) % Controller Derivative Time in Secs

Td = 0

Tf = Tdt*Lbd/(2*(Tdt+Lbd)) % Derivative Filter Time Constant in sec

Tf = 0
```