



# Controladores PID

## Controle de Processos Industriais (CPI)

Departamento de Engenharia de Controle e Automação  
Instituto de Ciência e Tecnologia – UNESP – Campus Sorocaba

**Prof. Dr. Dhiego Fernandes Carvalho**

dhiego.fernandes@unesp.br

# Objetivos

---

- Compreender a importância e as diversas aplicações dos controladores PID em processos industriais
- Definir e diferenciar os principais critérios de desempenho utilizados na avaliação de sistemas de controle
- Entender as funções e características das ações proporcional, integral e derivativa
- Distinguir entre as diferentes estruturas de controladores PID, incluindo as configurações em série, em paralelo e no modelo ISA, e suas implicações para o projeto de sistemas de controle.
- Reconhecer os efeitos da saturação dos atuadores na performance do controle.
- Compreender a necessidade de estratégias Anti-Windup, que evitam a acumulação excessiva da ação integral.
- Entender a importância da filtragem na ação derivativa, visando minimizar o impacto do ruído na derivada.
- Compreender o conceito de ponderação da referência e sua relevância para otimizar o desempenho do controlador em aplicações práticas.

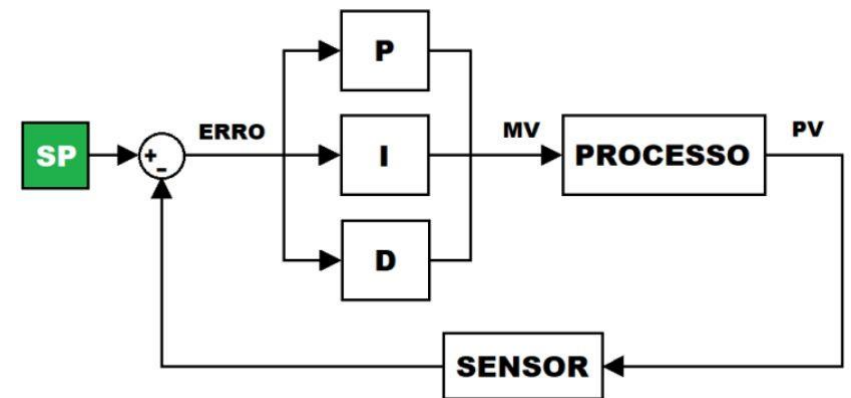
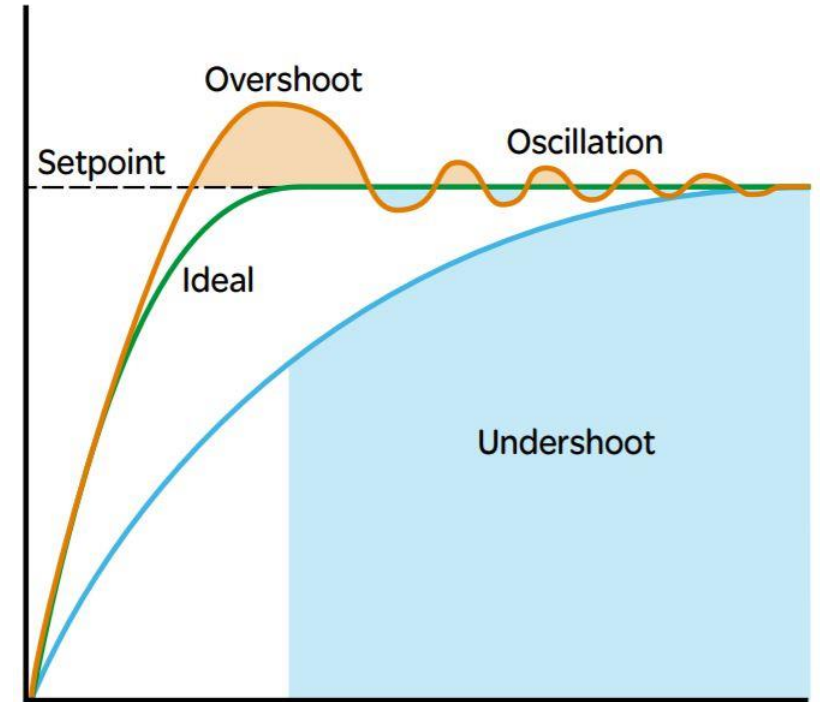
# Índice

---

- Introdução
- Critérios de Desempenho
- Componentes do PID
- Tipos de PID
- Saturação de atuadores e Anti-Windup
- Filtração da Ação Derivativa
- Ponderação de Referência
- Conclusões

# Introdução

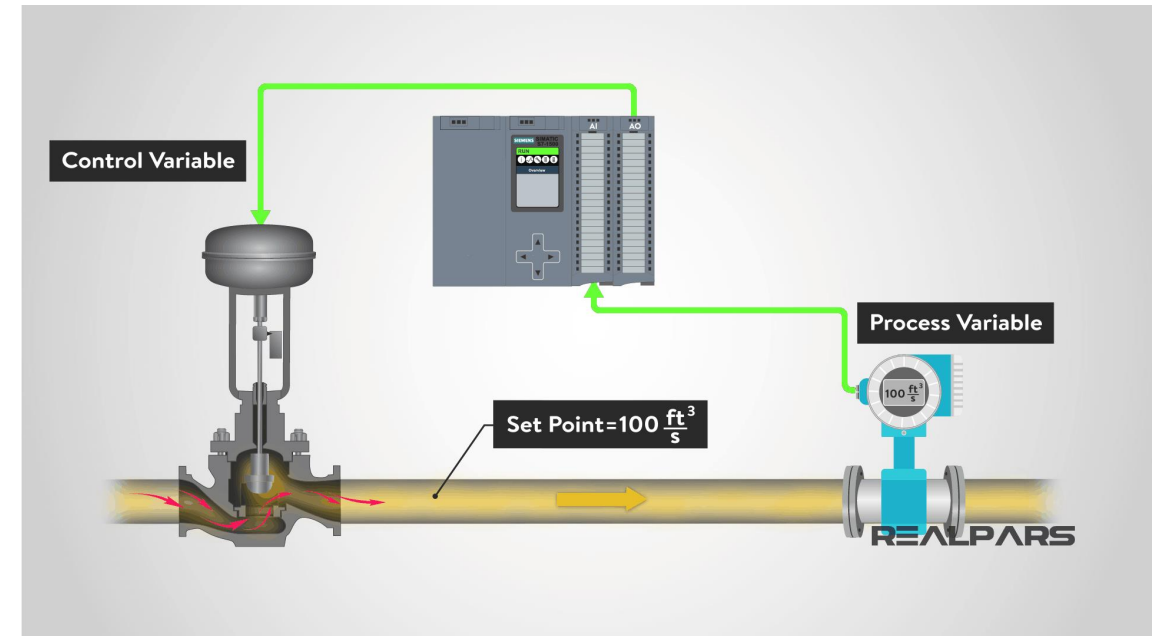
- **Controladores PID** (Proporcional-Integral-Derivativo) são um tipo de sistema de controle de feedback amplamente empregados em ambientes industriais.
- Sua principal função é regular variáveis de processo, mantendo-as o mais próximo possível de um valor desejado (*setpoint*).



# Introdução

---

- Os **controladores** são os pilares de muitas operações industriais, onde um pequeno erro ou atraso na resposta do sistema pode resultar em desperdício de recursos, tempo de inatividade ou, no pior dos casos, situações perigosas.
- Normalmente na indústria, o controladores PID são implementados nos **CLPs**.





# Introdução

- Os controladores PID são amplamente utilizados em diversos setores industriais.
- No farmacêutico, controlam variáveis como temperatura e umidade em salas limpas e processos de mistura;
- Na manufatura, controlam a velocidade de esteiras e a orientação de robôs;
- Na indústria de petróleo, gerenciam o controle de temperatura, pressão e níveis de líquidos em processos críticos.



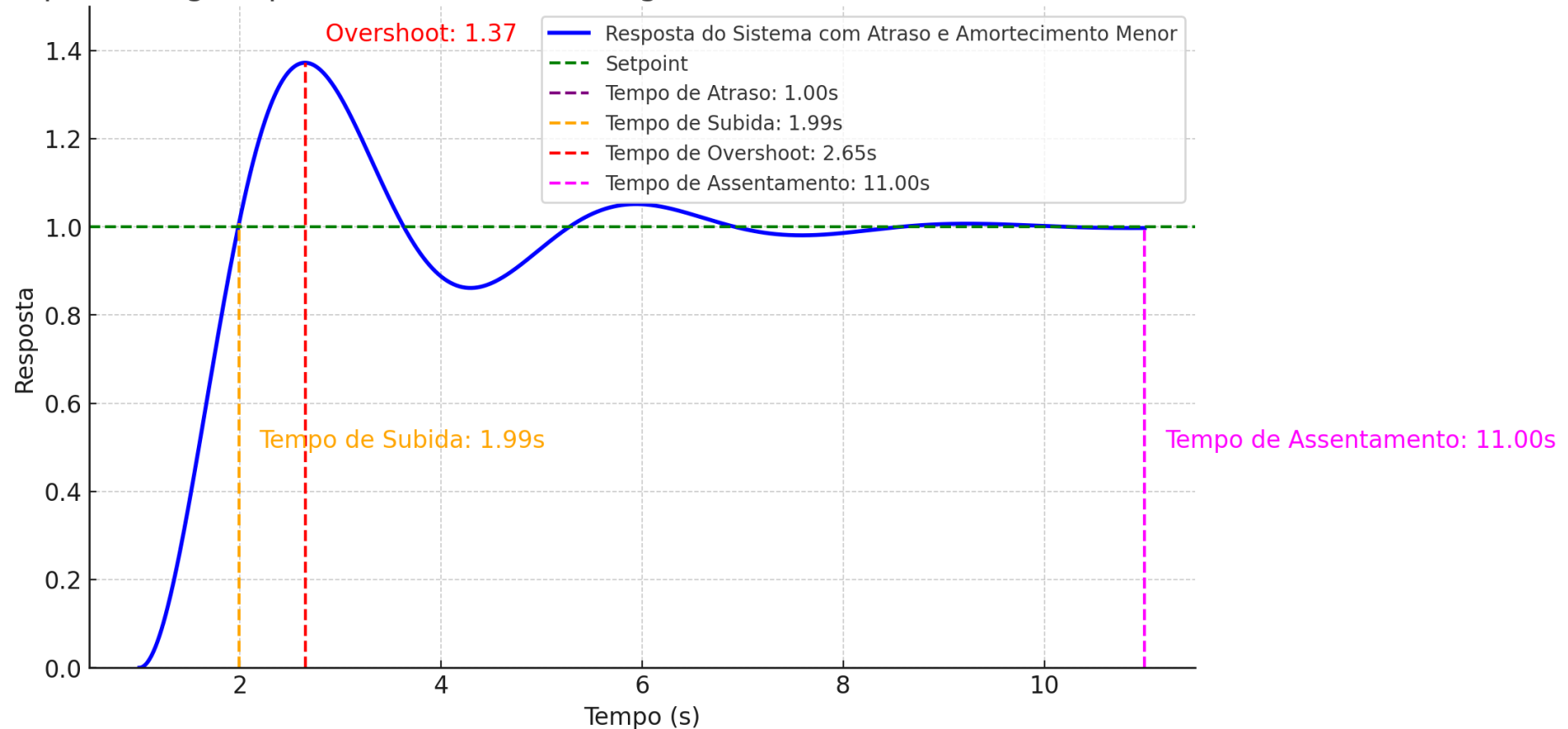
# Critérios de Desempenho

---

- Os critérios de desempenho em controladores PID são métricas que ajudam a avaliar o quão eficaz é um sistema de controle em alcançar suas metas.
  - **Tempo de Subida ( $T_r$ ):** O tempo de subida é o tempo que o sistema leva para ir de 10% a 90% (ou de 0% a 100%, dependendo da convenção) do valor de regime permanente a partir do momento em que um degrau é aplicado.
  - **Overshoot ( $M_p$ ):** O overshoot é o pico máximo da saída acima do valor de regime permanente.
  - **Tempo do Overshoot:** tempo para atingir o pico máximo do overshoot ( $M_p$ ).
  - **Erro em Regime Permanente ( $e_{ss}$ ):** O erro em regime permanente é a diferença entre a saída e o valor desejado após o sistema ter alcançado o estado estacionário.
  - **Tempo de Assentamento ( $T_s$ ):** Este é o tempo necessário para que a saída se estabilize dentro do  $e_{ss}$ .
  - **Tempo de Atraso (L):** É o atraso da resposta do sistema.

# Os critérios de desempenho ajudam nos controladores PID

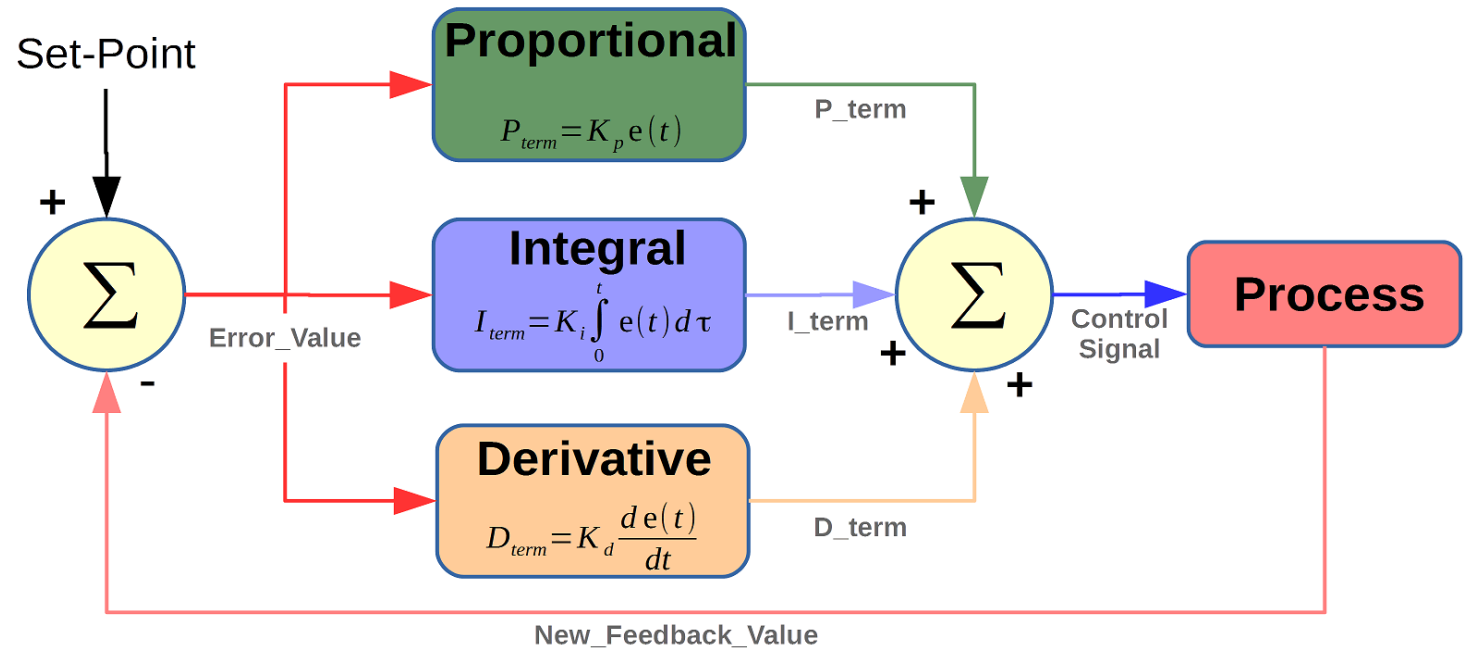
Resposta Degrau para um Sistema de Segunda Ordem Subamortecido com Atraso





# Componentes PID

- O controlador PID é composto por três ações fundamentais: **Proporcional (P)**, **Integral (I)** e **Derivativa (D)**. Essas ações trabalham juntas para controlar sistemas dinâmicos de diversas naturezas.



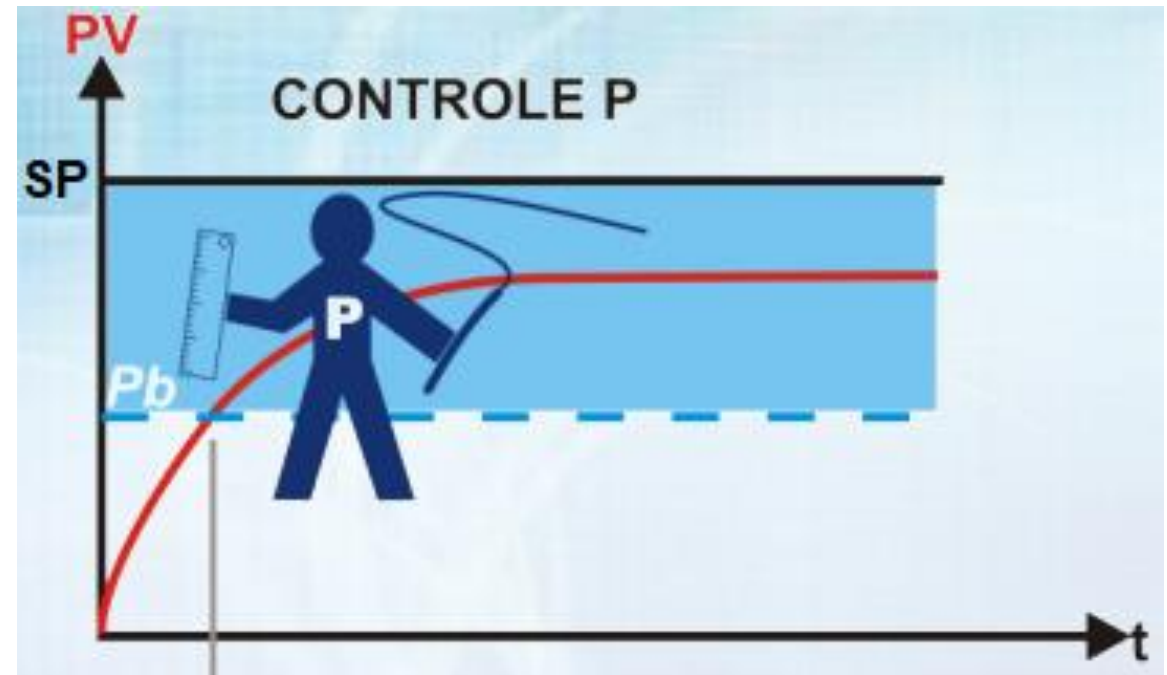
# Ação Proporcional

- É a mais simples entre as três. Ela produz uma saída que é proporcional ao erro atual  $e(t)$  entre o valor desejado (setpoint) e o valor atual da variável controlada.

$$u(t) = K_p \cdot e(t)$$

Onde:

- $K_p$ : Ganho Proporcional
- $e(t)$ : Erro = (Setpoint - Valor atual)



- Um aumento no ganho proporcional ( $K_p$ ) resulta em uma resposta mais rápida, mas pode causar **oscilações** e **ultrapassagens**.
- Em sistemas somente com ação proporcional, é comum que haja um pequeno erro de regime permanente, já que o controlador proporcional não pode eliminar o erro por completo.

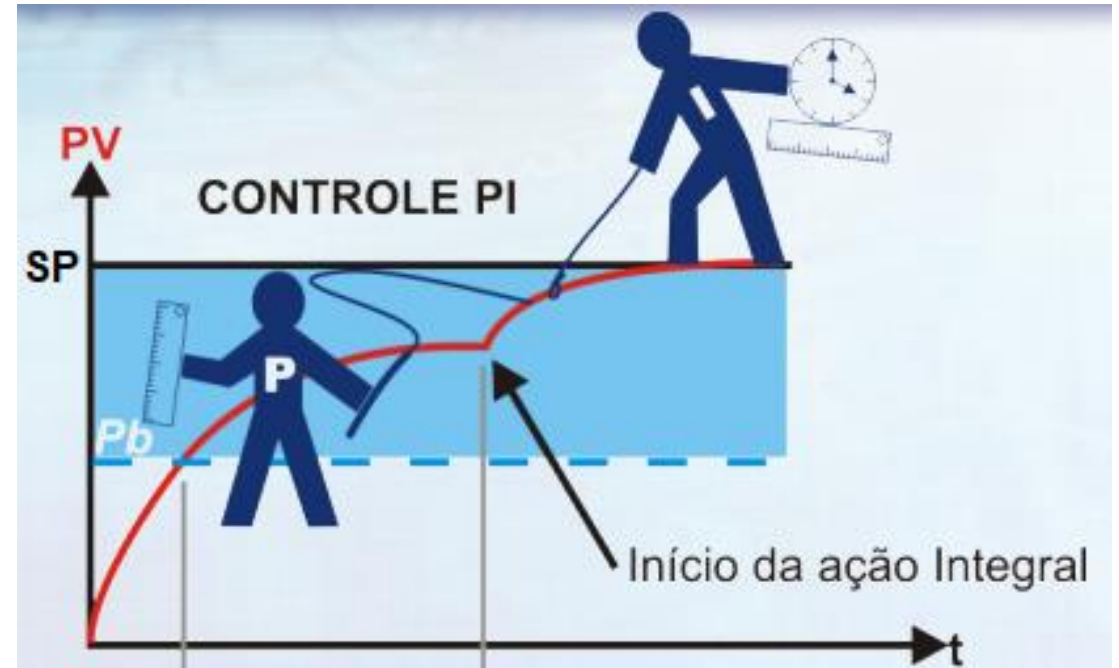
# Ação Integral

- Busca eliminar o erro acumulado ao longo do tempo, agindo sobre a integral do erro.

$$u(t) = K_i \int_0^t e(\tau) d\tau$$

Onde:

- $K_i$ : Ganho Integral



- Quando a ação Integral é aplicada isoladamente, ela pode comprometer a estabilidade do sistema.
- No entanto, quando combinada com a ação Proporcional, a ação Integral ajuda a eliminar o erro de regime permanente.
- Quanto maior o valor do ganho Integral, mais tempo o sistema pode levar para alcançar o setpoint (SP) de forma estável, pois um valor elevado pode causar oscilações prolongadas.

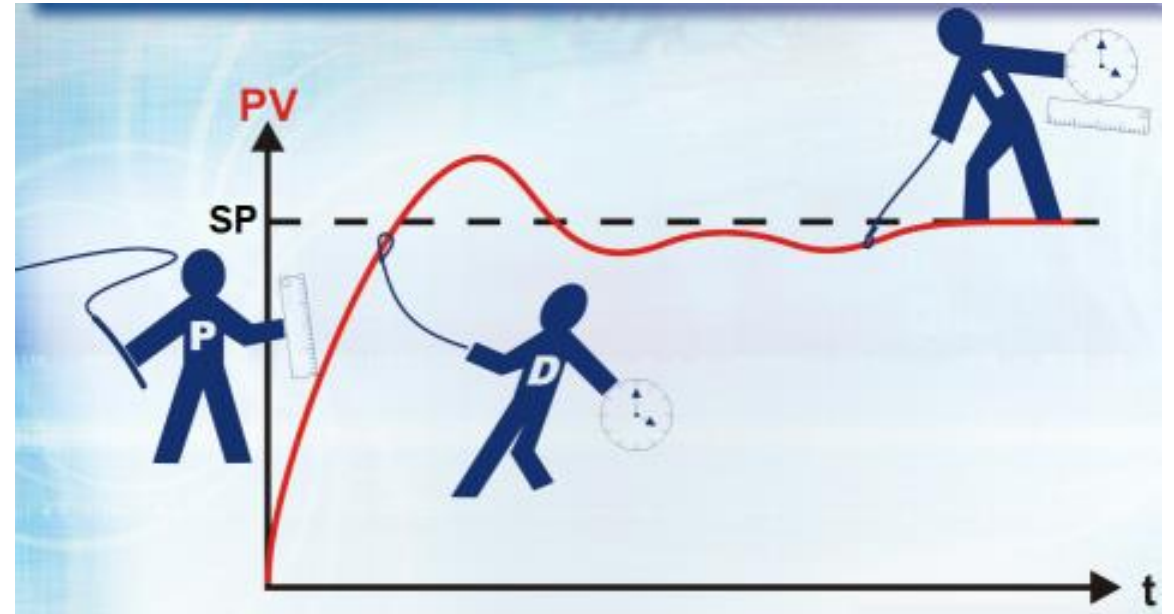
# Ação Derivativa

- Atua sobre a taxa de variação do erro, tentando prever o comportamento futuro do sistema e agindo de forma preventiva.

$$u(t) = K_d \cdot \frac{de(t)}{dt}$$

Onde:

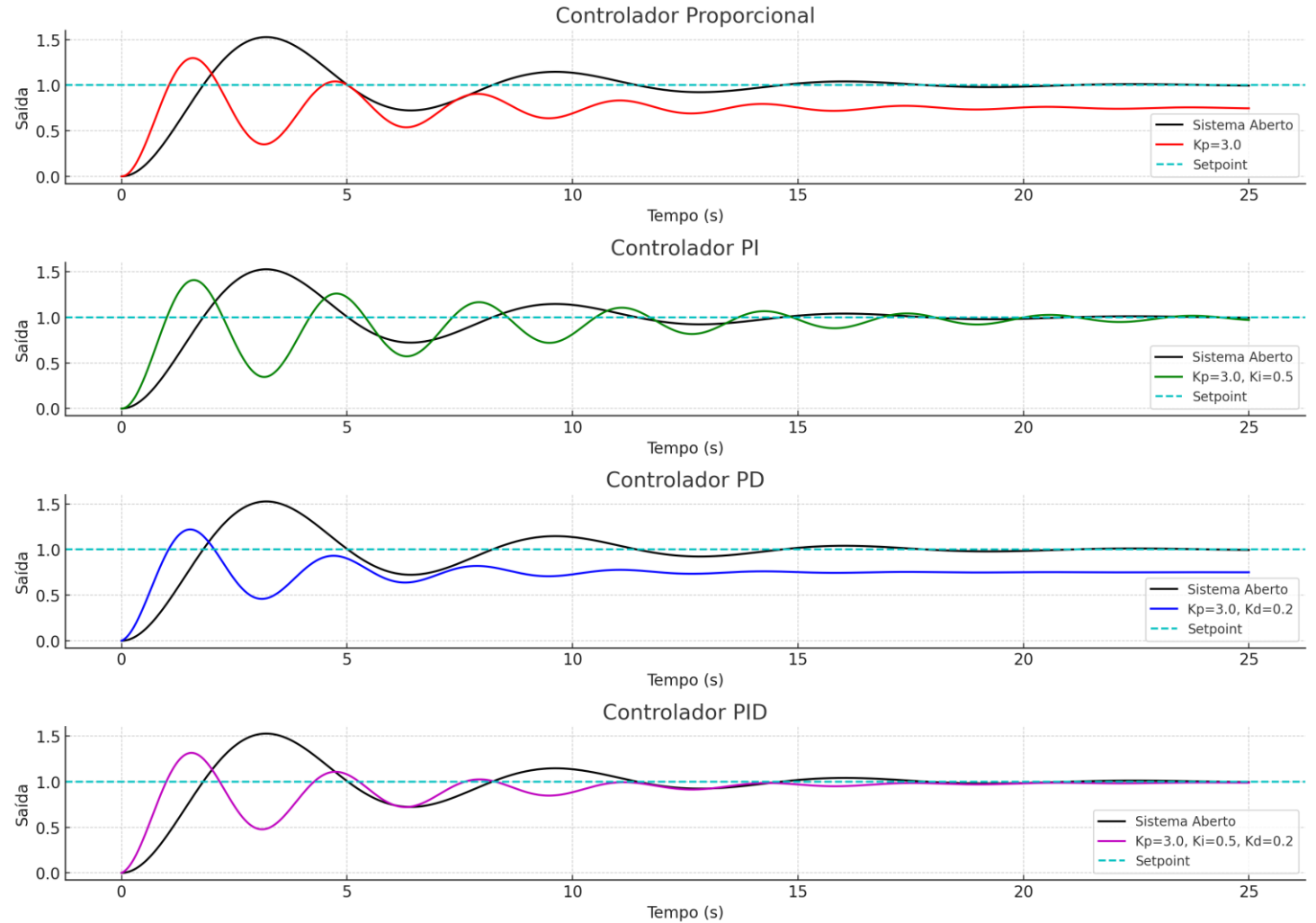
- $K_d$ : Ganho Derivativo



- A ação Derivativa antecipa a ação de controle, permitindo que o processo reaja mais rapidamente do que normalmente faria.
- Essa ação preditiva aumenta a estabilidade relativa do sistema, tornando a resposta mais rápida e menos oscilatória à medida que o tempo derivativo é ajustado.
- Em regime permanente, a ação Derivativa se torna nula, já que o erro permanece constante.

# Ações Proporcionais, Integrais e Derivativas

- **Ação Proporcional (P):** Aumenta a velocidade de resposta do sistema, mas pode deixar um erro estacionário (*offset*) em regime permanente.
- **Ação Integral (I):** Elimina o erro estacionário ao acumular o erro ao longo do tempo e corrigi-lo. No entanto, pode reduzir a estabilidade do sistema, tornando-o mais suscetível a *overshoot* e oscilações.
- **Ação Derivativa (D):** Prevê a tendência futura das oscilações e atua para minimizá-las, aumentando a estabilidade do sistema e reduzindo o *overshoot*. Porém, é sensível ao ruído do sinal, o que pode comprometer seu desempenho.



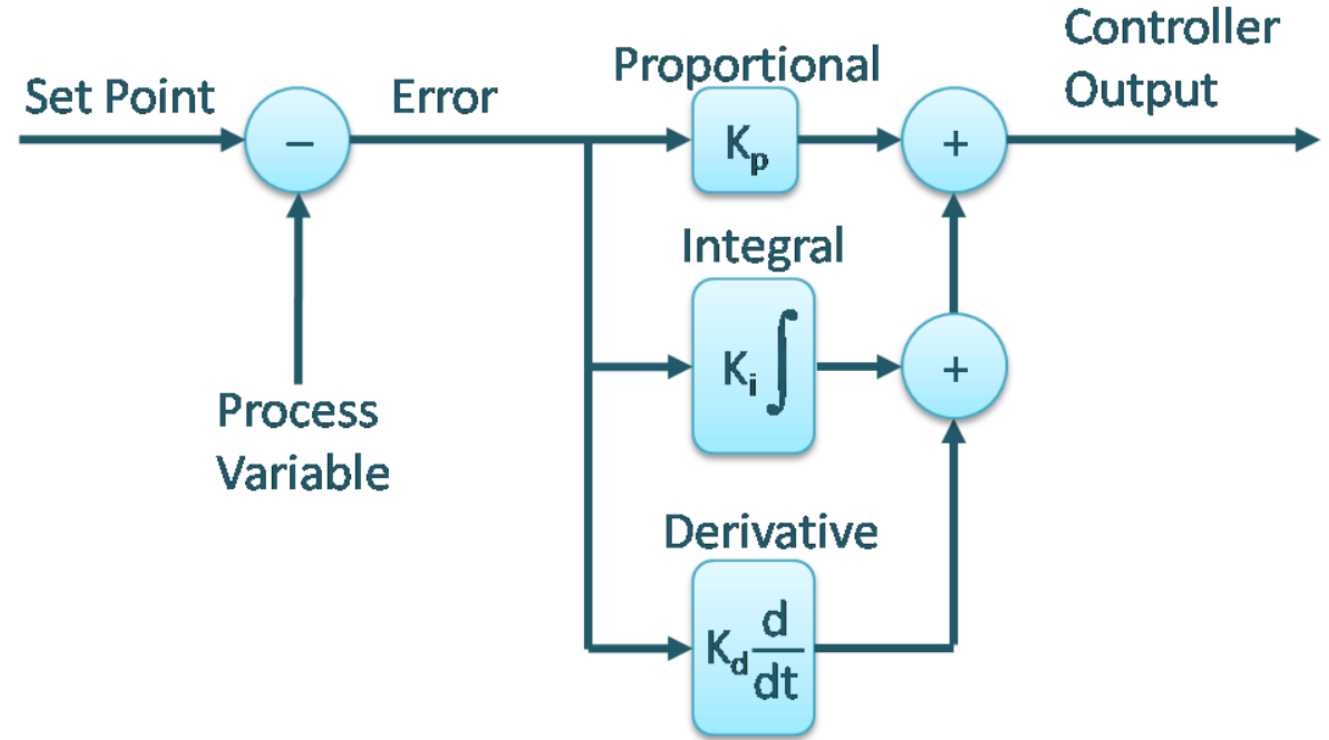
# Controlador PID em Paralelo

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

$$G(s) = K_p + \frac{K_i}{s} + K_d s$$

Onde:

- $K_p$  é o ganho proporcional.
- $T_i$  é a constante de tempo integral ( $T_i = \frac{1}{K_i}$ ), sendo  $K_i = \frac{1}{T_i}$ .
- $T_d$  é a constante de tempo derivativa ( $T_d = K_d$ ), sendo  $K_d = T_d$ .
- $\tau$  é a constante de tempo do filtro para a ação derivativa



- Fácil de entender e implementar.
- Os termos PID podem ser ajustados de forma independente.
- Pode não fornecer um desempenho ideal para todos os sistemas devido à falta de interação entre os termos PID.

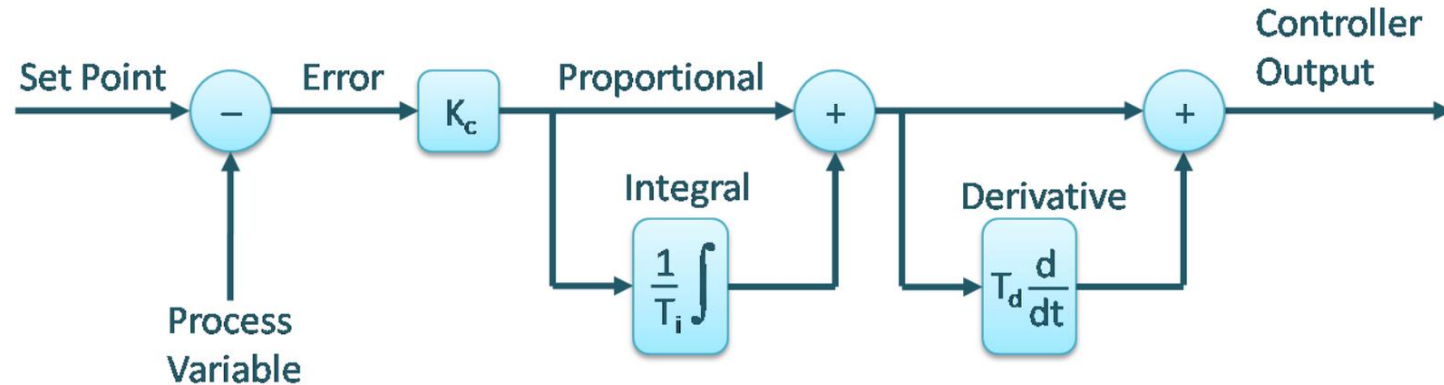
# Controlador PID em Série (interativo)

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau \right] \cdot \left[ 1 + T_d \frac{de(t)}{dt} \right]$$

$$G(s) = K_p \left( 1 + \frac{1}{T_i s} \right) (1 + T_d s)$$

Onde:

- $K_p$  é o ganho proporcional.
- $T_i$  é a constante de tempo integral  $\left( T_i = \frac{K_p}{K_i} \right)$ , sendo  $K_i = \frac{K_p}{T_i}$ .
- $T_d$  é a constante de tempo derivativa  $\left( T_d = \frac{K_d}{K_p} \right)$ , sendo  $K_d = K_p T_d$ .
- $\tau$  é a constante de tempo do filtro para a ação derivativa



- O **controlador PID interativo** é chamado assim porque suas ações não atuam de maneira independente, mas **interagem entre si**, ou seja, as ações Integral e Derivativa influenciam diretamente a ação Proporcional, criando um comportamento encadeado.



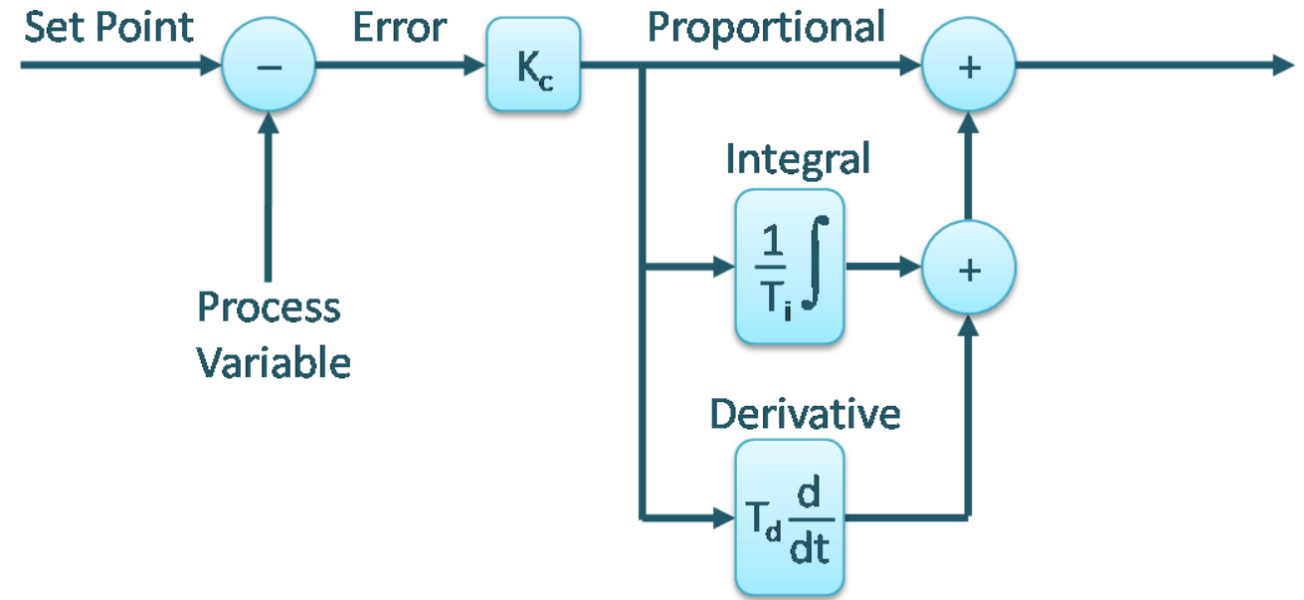
# Controlador PID modelo ISA

$$u(t) = K_p \left[ e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right]$$

$$G(s) = K_p + \frac{K_p}{T_i s} + K_p T_d s$$

Onde:

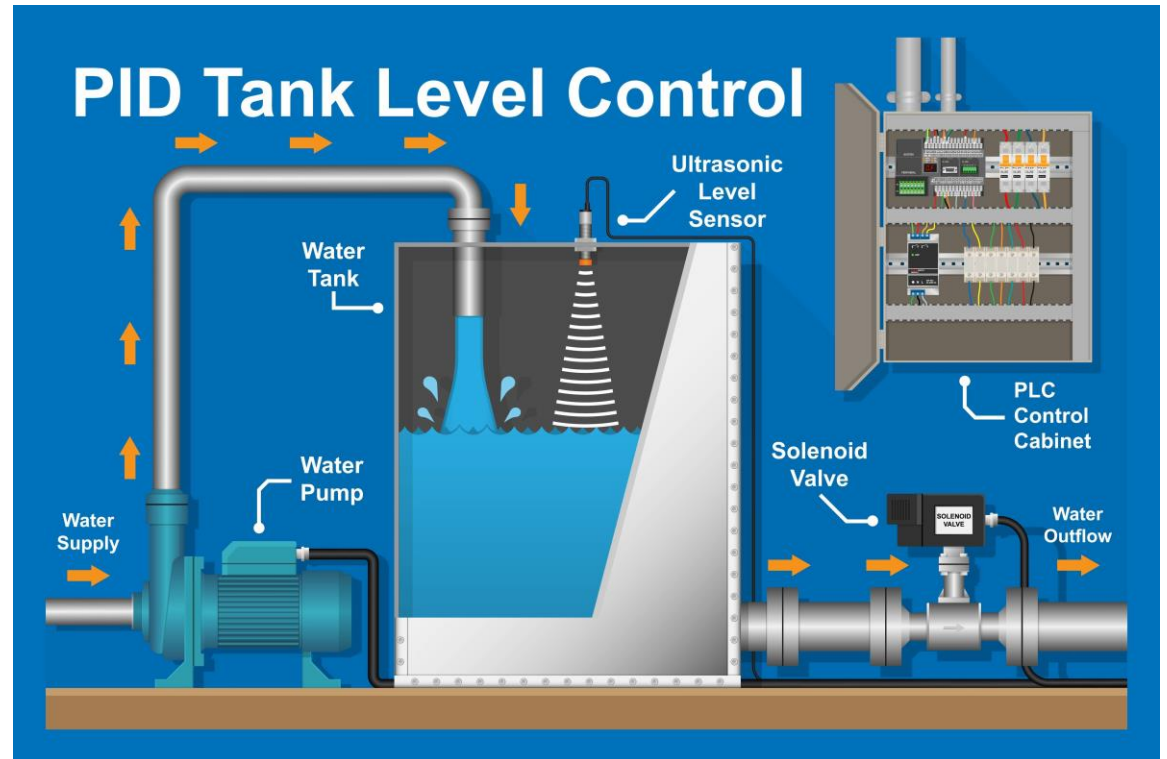
- $K_p$  é o ganho proporcional.
- $T_i$  é a constante de tempo integral  $\left(T_i = \frac{K_p}{K_i}\right)$ , sendo  $K_i = \frac{K_p}{T_i}$ .
- $T_d$  é a constante de tempo derivativa  $\left(T_d = \frac{K_d}{K_p}\right)$ , sendo  $K_d = K_p T_d$ .
- $\tau$  é a constante de tempo do filtro para a ação derivativa.



- Proporciona um equilíbrio entre a facilidade de sintonia do controlador paralelo e a interação dos termos PID do controlador em série.
- Adequado para uma ampla gama de sistemas.

# Tipos de PID

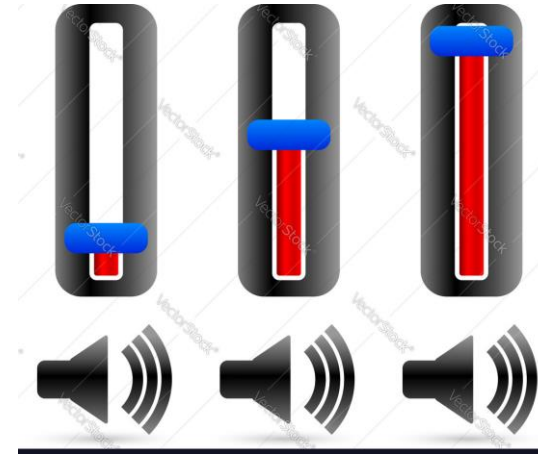
- **Controlador PID em Paralelo:** é útil quando é desejável ajustar as ações Proporcional, Integral e Derivativa de forma independente, permitindo maior flexibilidade no ajuste dos parâmetros de controle.
- **Controlador PID em Série (ou Interativo):** é usado em cenários onde a interação entre as ações de controle é desejada, mas pode ser mais difícil de ajustar em comparação com o PID em paralelo.
- **Controlador PID no Modelo ISA:** é um padrão na indústria e é frequentemente usado em sistemas que exigem alta confiabilidade e robustez.



# Saturação de Atuadores

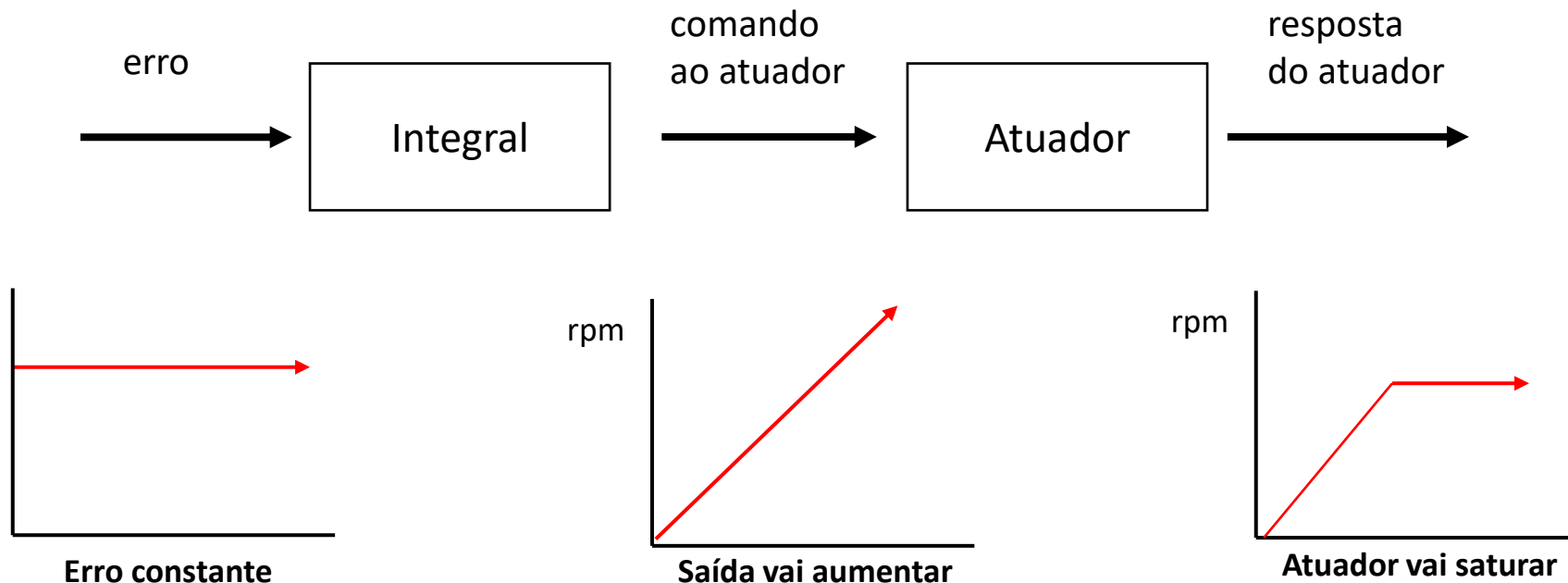
---

- A **saturação de atuadores** acontece quando o atuador atinge seus limites operacionais.
- Um motor pode não conseguir girar mais rápido do que uma certa RPM, uma bateria não pode fornecer uma amperagem acima da sua capacidade e um alto-falante apenas pode fornecer um volume até um certo nível.



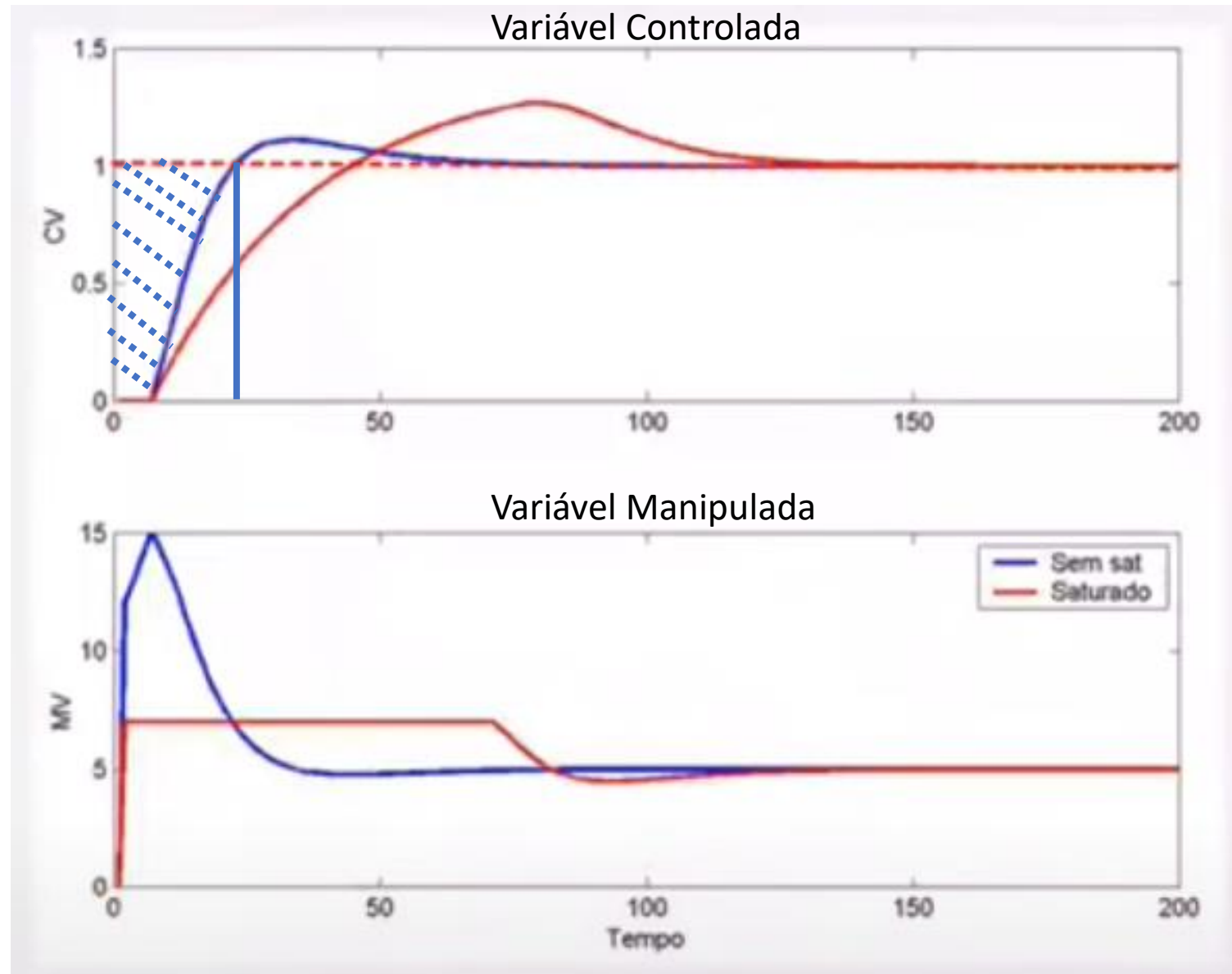
# Efeito *windup*

- A saturação dos atuadores é frequentemente um problema que ocorre devido à **ação integral** do controlador PID.
- Em um cenário ideal, o termo integral ajuda a eliminar o erro estacionário, fazendo ajustes cumulativos com base no histórico do erro.
- No entanto, na presença de limitações de atuador, a ação integral pode continuar a acumular, levando a uma saída de controle que excede os limites do atuador. Esse fenômeno é conhecido como **windup**.



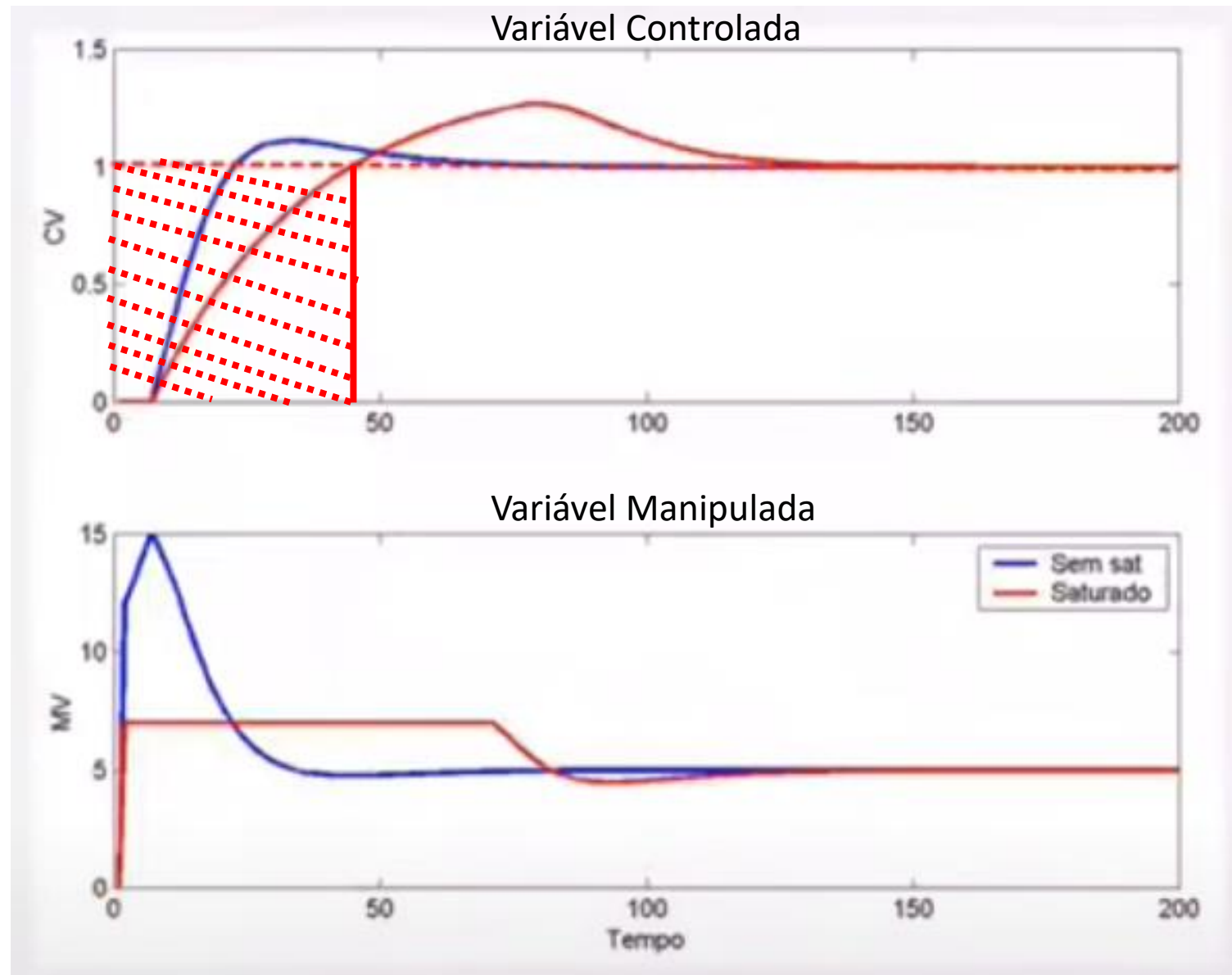
## Efeito *Windup*

- Em um controle PI ou PID sem saturação, o *overshoot* é menor. Devido a diferença entre a variável controlada e o setpoint ser menor.



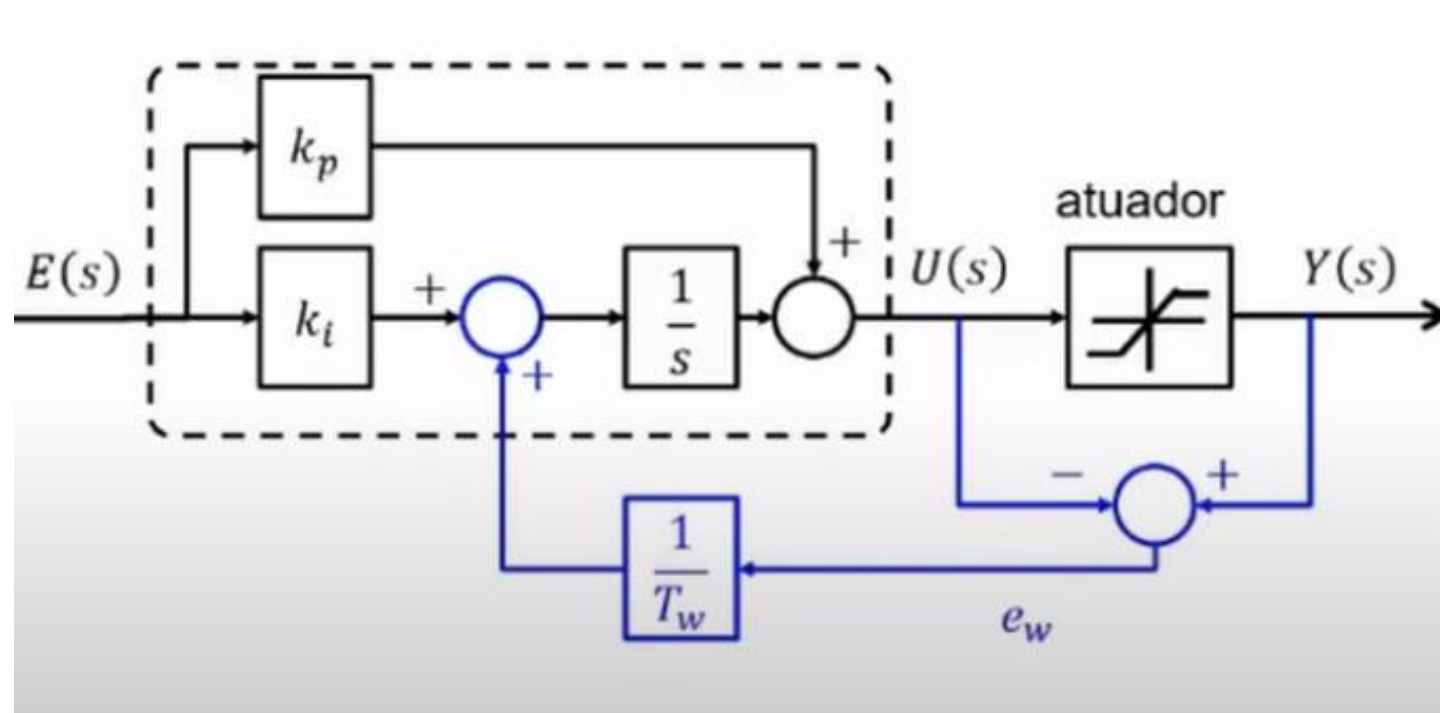
## Efeito *Windup*

- Em um controle PI ou PID com saturação, o *overshoot* é maior. Devido a diferença entre a variável controlada e o setpoint ser maior.



# Métodos *Anti-Windup*

- Para se evitar o *Windup*, existem métodos *anti-Windup*. Um deles é o *Back-Calculation*. Quando o atuador atinge seu limite, a ação integral é recalculada na forma que seu valor permaneça no limite de sua atuação.
- Esse ajuste não é feito imediatamente, mas dinamicamente com uma constante de tempo  $T_w$ .

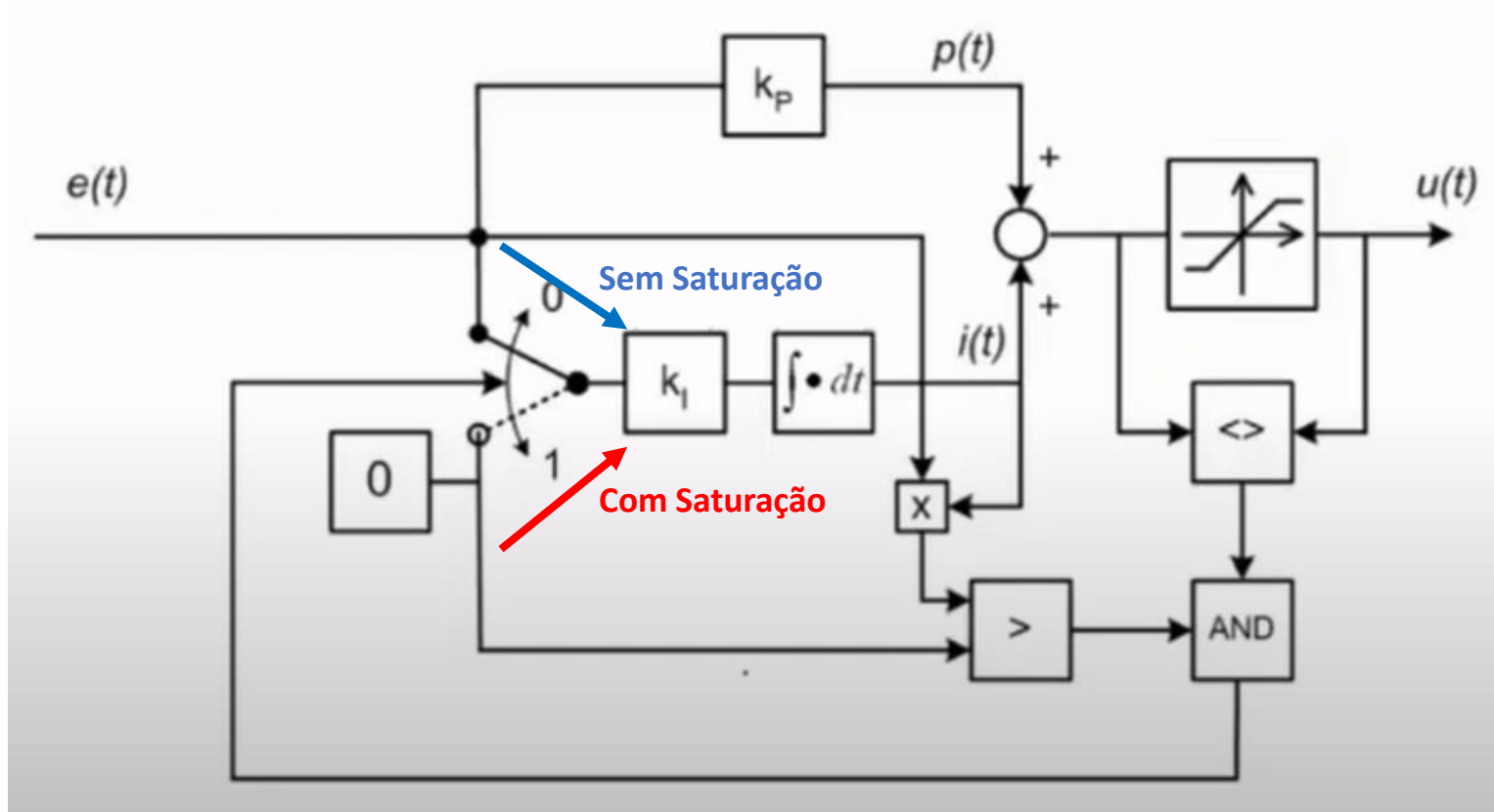


Uma boa métrica para obter  $T_w = \sqrt{T_i T_d}$  ou  $T_w = \frac{T_i}{K_p}$



## Métodos Anti-Windup

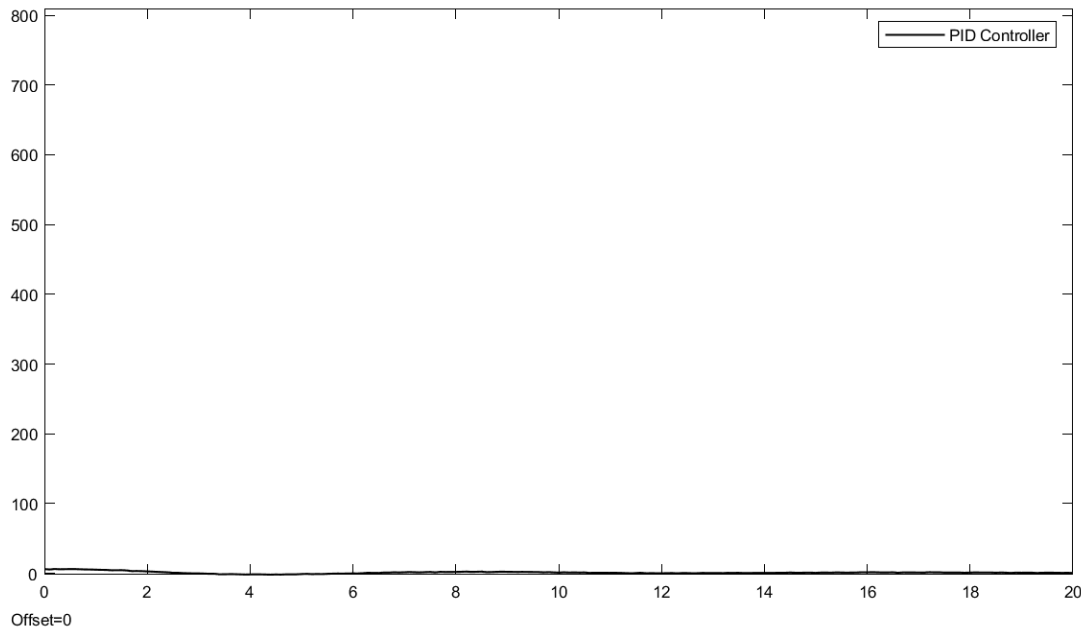
- A **Integração Condicional** é outra técnica *anti-Windup*. Ela é usada para desligar a ação integral do controlador quando o atuador saturar.



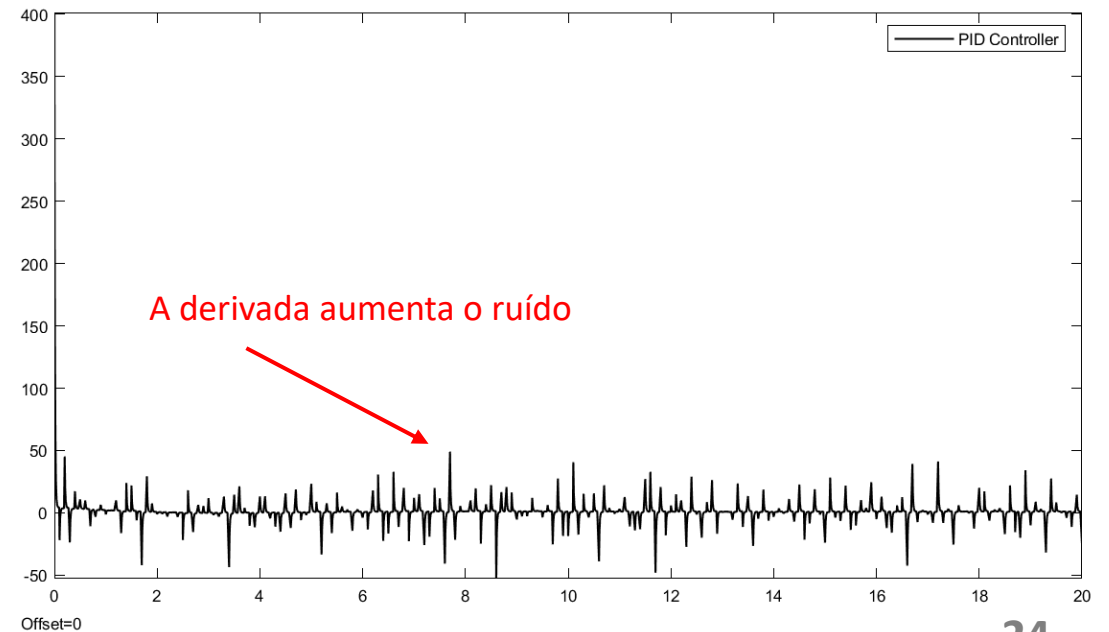
# Filtragem de Ação Derivativa

- É uma técnica usada para suavizar a ação derivativa do controlador, mitigando o impacto de ruídos e oscilações bruscas no sinal de entrada.
- A filtragem de Ação Derivativa vem por padrão nos controladores PID tipo ISA.

Saída do PID em um sistema com ruído, mas a ação derivativa desligada

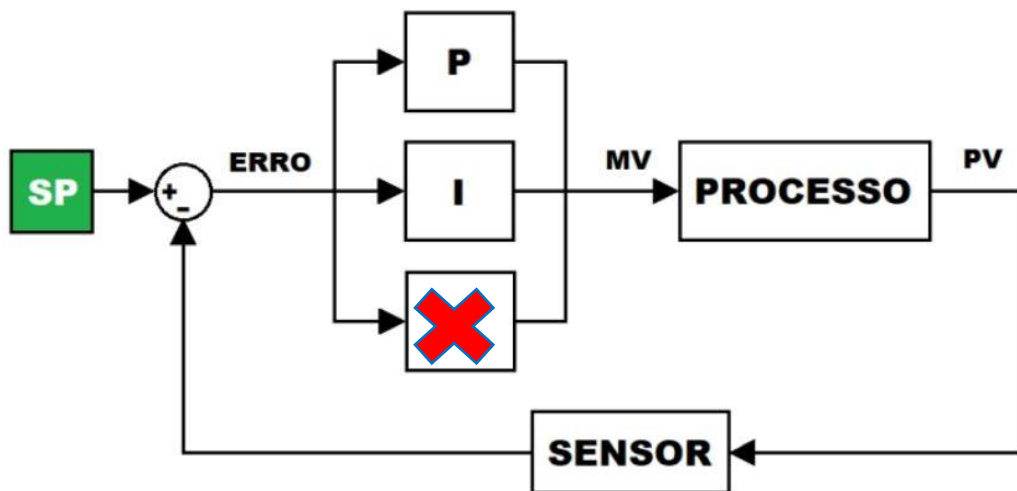


Saída do PID em um sistema com ruído



# Filtragem de Ação Derivativa

- A filtragem da ação derivativa geralmente envolve passar o sinal de erro através de um filtro de baixa passagem antes de calcular a ação derivativa.
- O filtro de baixa passagem pode ser um filtro simples de primeira ordem, um filtro PID com apenas a ação proporcional e integral ativas, ou outro tipo de filtro que atenua as frequências mais altas do sinal.



# Filtragem de Ação Derivativa

---

- A função de transferência da ação derivativa filtrada pode ser expressa da seguinte maneira:

$$D(s) = \frac{T_f s}{\frac{T_f s}{N} + 1} = K_d \left( \frac{N}{1 + \frac{N}{s}} \right)$$

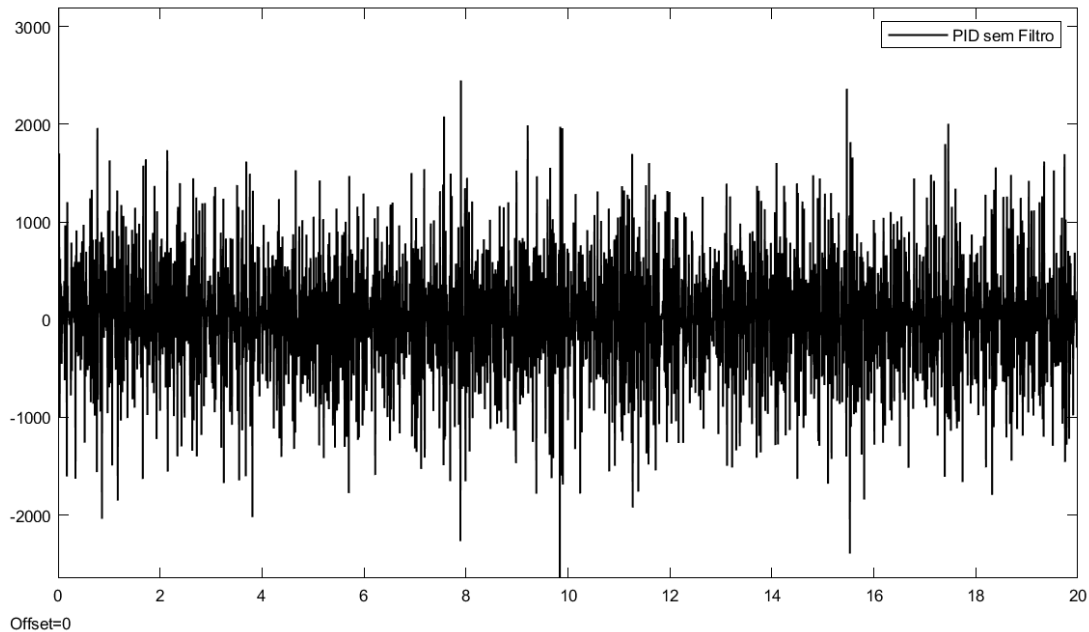
Onde:

- $T_f$  é a constante de tempo do filtro derivativo, que determina a largura da banda do filtro  $T_f = \frac{T_d}{N}$
- $N$  determina a frequência de corte do filtro. Um valor maior de  $N$  resulta em uma frequência de corte mais alta, permitindo que mais componentes de alta frequência passem.
- **Balanceamento:** um  $N$  alto pode não filtrar o ruído suficientemente, enquanto um  $N$  baixo pode fazer com que a ação derivativa responda lentamente às mudanças no erro (normalmente valor  $N$  varia entre 1 a 20, dependendo do sistema).

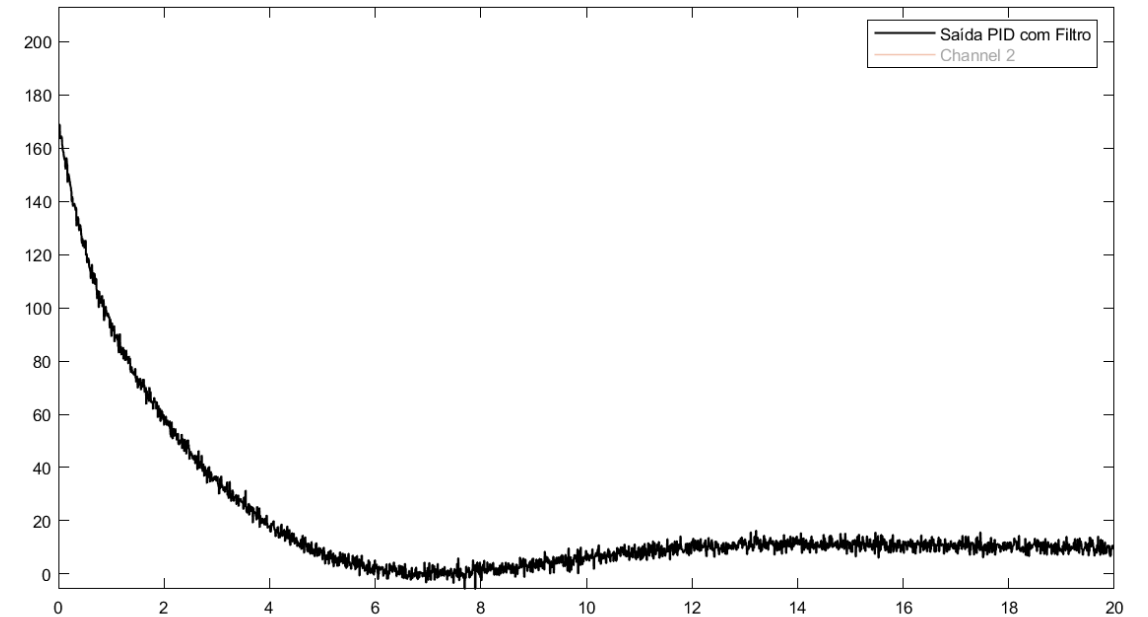
# Filtragem de Ação Derivativa

- A filtragem da ação derivativa ajuda a suavizar essas oscilações de ruído, resultando em um controle mais estável e eficaz.

Saída do PID sem Filtragem

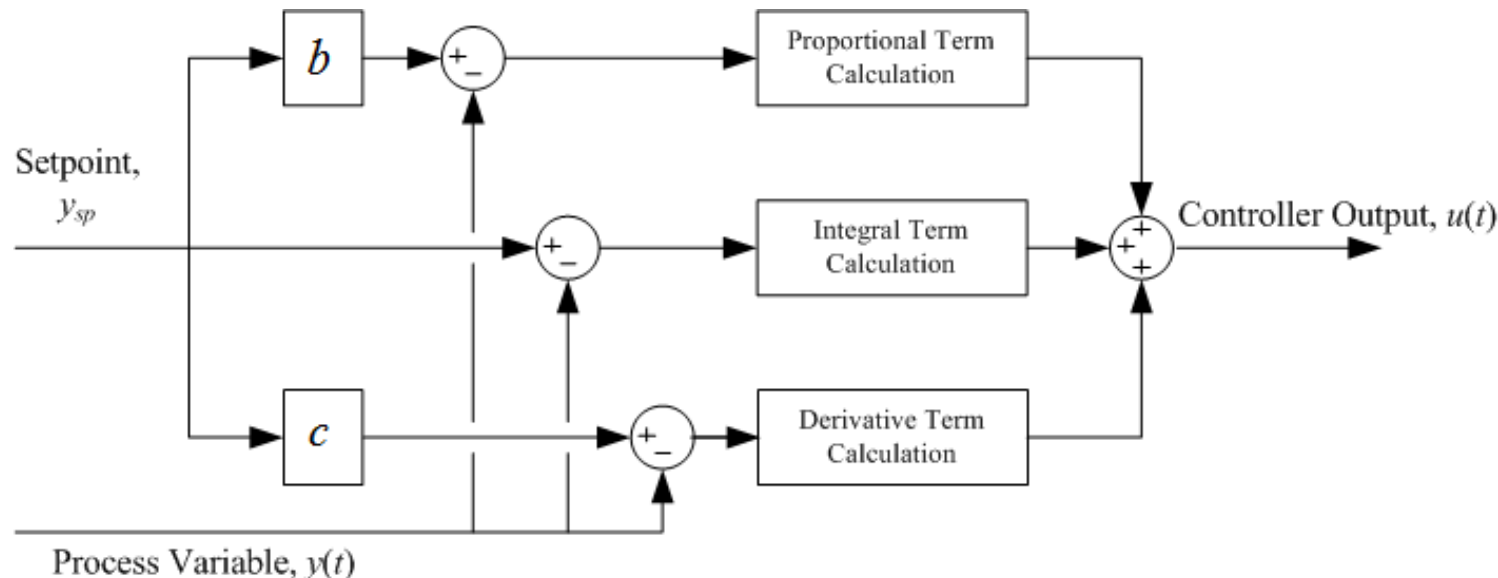


Saída do PID com filtragem



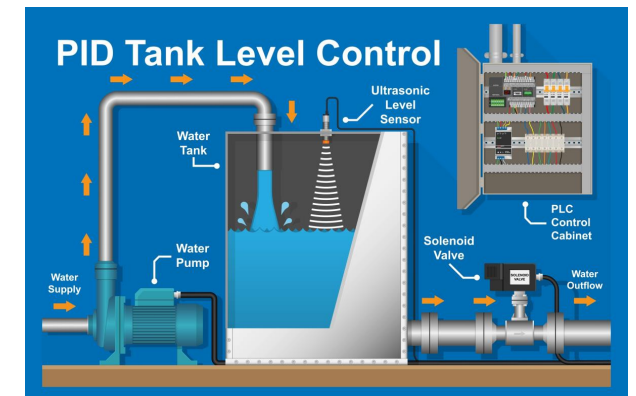
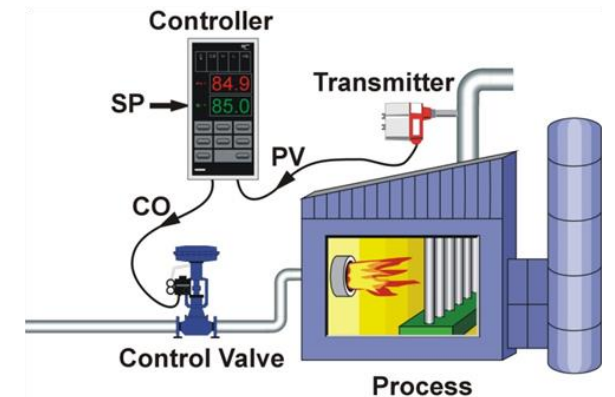
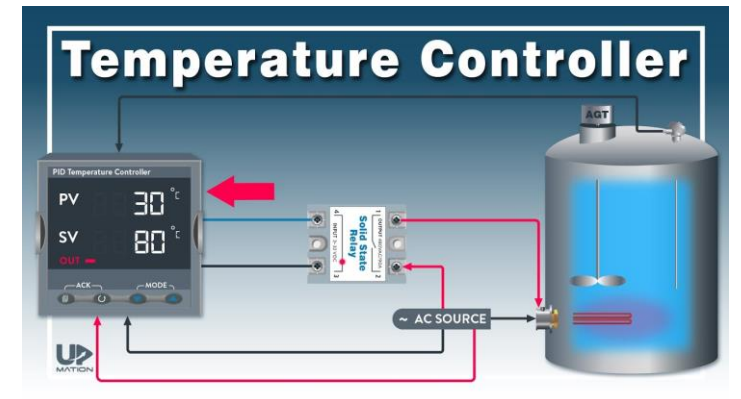
# Ponderação de Referência

- A **ponderação de referência**, também conhecida como ponderação de **setpoint**, é uma técnica usada em sistemas de controle para melhorar a resposta transitória do sistema quando o *setpoint* (referência) muda.
- Isso é feito ajustando como o controlador responde a mudanças no setpoint em comparação com as mudanças no erro de rastreamento.



# Onde é usada a Ponderação de Referência?

- A ponderação de referência é usada em **controles de temperatura de Fornos Industriais, Sistema de Controle de Processos Químicos, Controle de Nível de Tanque, Controle de Velocidade de Motores, etc.**
- Resumindo: ela é usada em uma ampla variedade de sistemas onde mudanças rápidas no *setpoint* podem causar oscilações, ultrapassagens ou respostas exageradas do controlador.

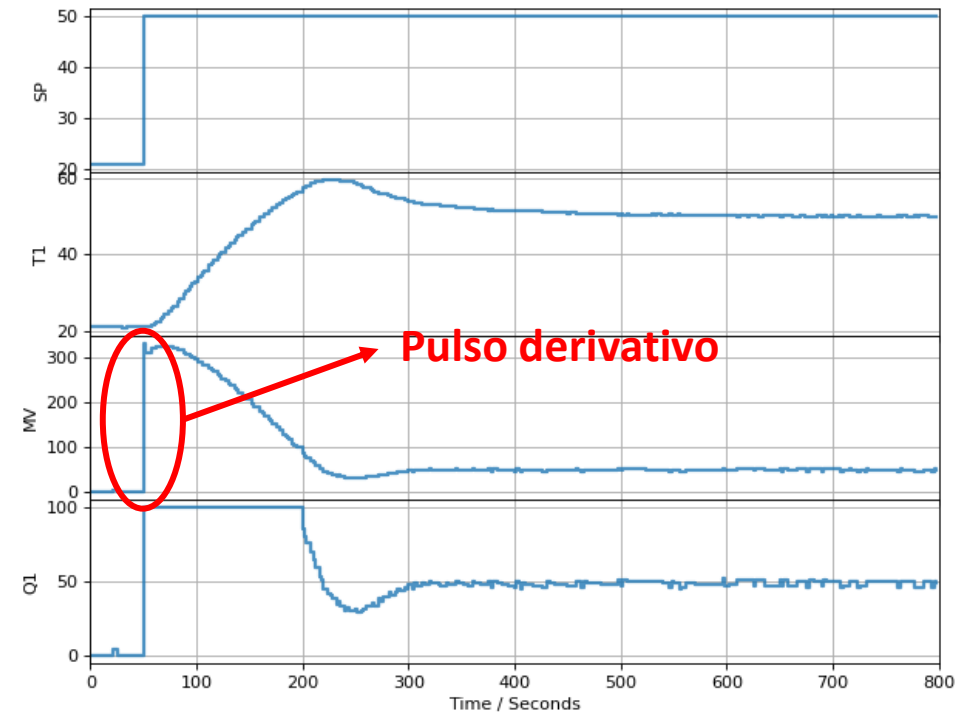




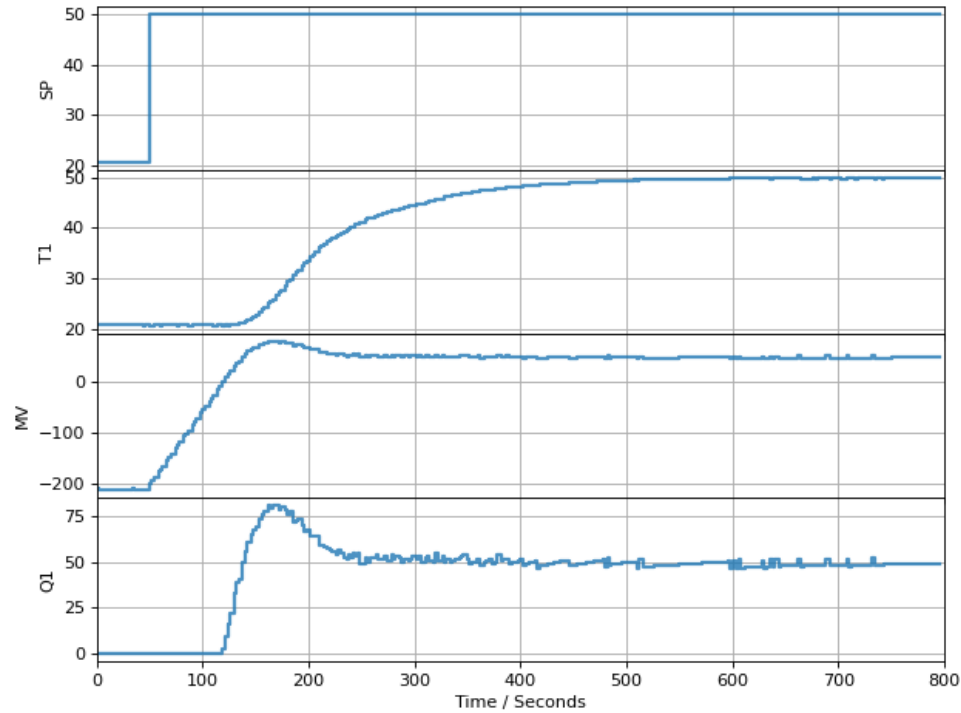
# O que é o pulso derivativo?

- Se o setpoint mudar de forma brusca a taxa de variação do erro também mudará instantaneamente, o que pode levar a um **pico muito grande** na saída da ação derivativa, resultando em uma resposta exagerada do controlador.

## Sistema de Controle de Temperatura: mudança de setpoint de 20°C para 50°C



Sem Ponderação de Referência



Com Ponderação de Referência

**SP:** Setpoint  
**T1:** Variável de Saída do Processo  
**MV:** Variável Manipulada pelo PID  
**Q1:** Potência aplicada pelo aquecedor

# Fórmula da Ponderação de Referência

---

- O controlador PID padrão é expresso pela equação:

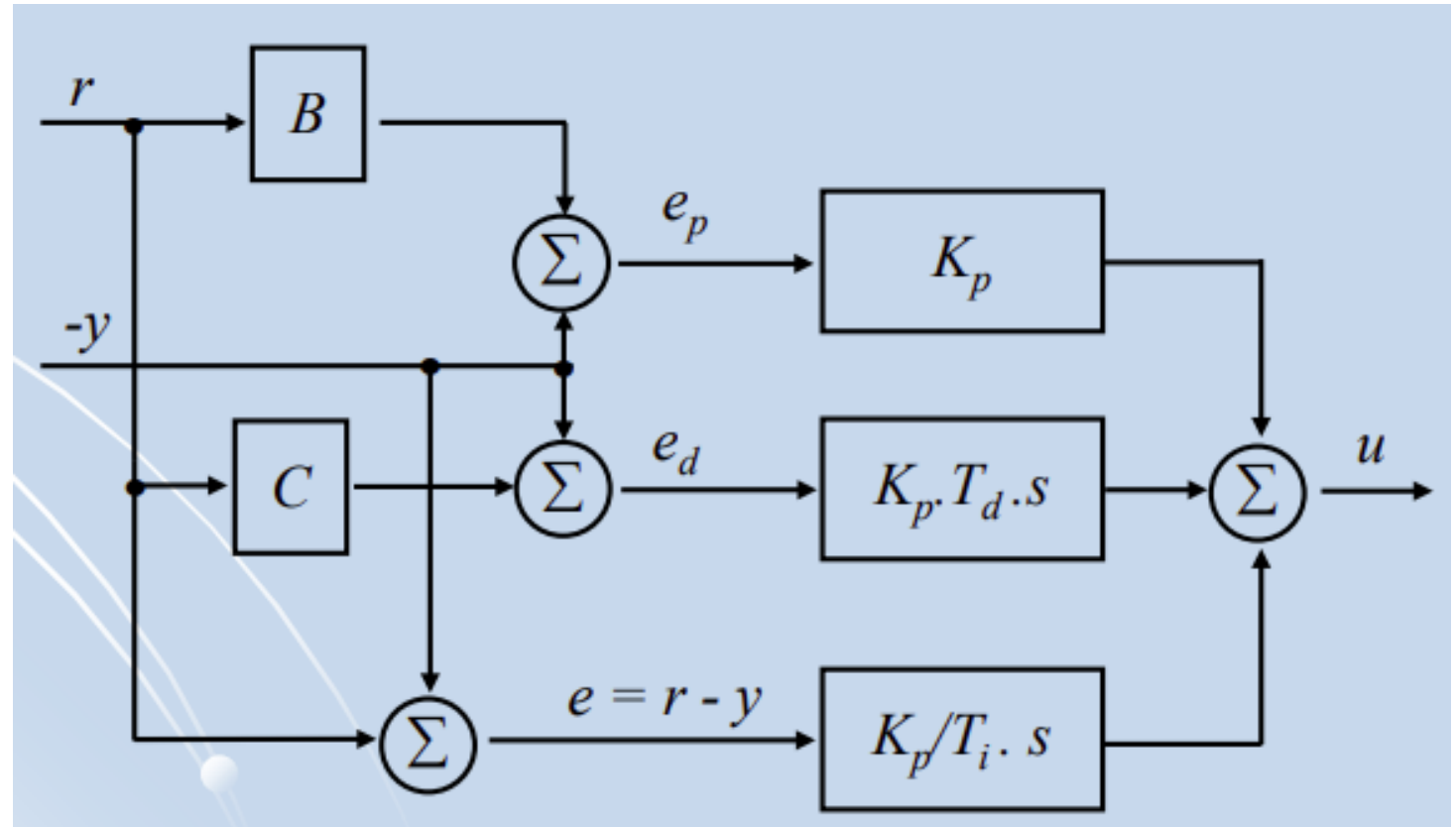
$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt}$$

- Onde  $u(t)$  é a saída do controlador,  $e(t) = r(t) - y(t)$  é o erro entre o *setpoint*  $r(t)$  e a saída do processo  $y(t)$ .
- Na ponderação de referência (*setpoint*), os termos proporcional e derivativo do controlador PID são modificados para incluir pesos diferentes (**variáveis  $b$  e  $c$** ) para o setpoint e para o erro de medida. A equação do controlador torna-se:

$$u(t) = K_p (b \cdot r(t) - y(t)) + K_i e(t) + K_d \left( c \cdot \frac{dr(t)}{dt} - \frac{dy(t)}{dt} \right)$$

# Tipos de Ponderação de Referência

- **Ponderação Proporcional ( $b$ ):** A ponderação proporcional afeta diretamente o quanto o controlador responde a mudanças no setpoint. Um valor de  $b$  maior resultará em uma resposta mais agressiva a mudanças no setpoint.
- **Ponderação Derivativa ( $c$ ):** Às vezes, uma ponderação derivativa também pode ser aplicada, embora isso seja menos comum.



# Conclusões

---

- Nesta aula, exploramos o funcionamento dos controladores PID e suas aplicações, com foco em seus componentes principais: **Proporcional**, **Integral** e **Derivativo**. Também discutimos os diferentes tipos de implementação do PID, incluindo as formas em **série**, **paralelo** e o **modelo ISA**.
- Abordamos como a saturação do atuador pode afetar negativamente a ação integral, levando ao efeito de **windup**. Para mitigar esse problema, foram apresentadas as técnicas de **anti-windup**, que ajudam a manter a estabilidade e o desempenho do controlador.
- Demonstramos também como a ação derivativa pode amplificar ruídos no sistema, e ensinamos como **técnicas de filtragem derivativa** podem ser aplicadas para suavizar a resposta do controlador, melhorando sua robustez.
- Por fim, explicamos **as técnicas de ponderação da referência**, que permitem otimizar a resposta do controlador PID em mudanças de *setpoint*, melhorando o comportamento do sistema em transições.

# DÚVIDAS?

---

# Exercícios

---

## 1. Circuitos com Controladores PID:

- Crie **quatro circuitos em malha fechada no Simulink**, utilizando os seguintes tipos de **controladores PID**:
  - PID em **série**,
  - PID em **paralelo**,
  - PID no **formato ISA**,
  - e o **bloco padrão PID** do Simulink.
- Realize a **sintonia manual** em cada controlador e compare as **diferenças nas saídas** de cada circuito.

## 2. Controlador PI com Saturação:

- Usando o mesmo processo, crie um circuito em malha fechada com um **controlador PI**.
- Adicione uma **saturação no atuador** e observe como ela afeta o desempenho.
- Em seguida, duplique esse circuito, mas aplique uma técnica de **anti-windup** no controlador PI.
- Compare as **saídas dos dois circuitos**, observando o comportamento com e sem **anti-windup**.

## 3. Controlador PID com Ruído e Filtragem Derivativa:

- Usando o mesmo processo, crie um circuito com um **controlador PID no formato ISA**.
- Aplique um **ruído branco** ao sistema e observe o efeito na saída.
- Crie outro circuito idêntico, mas desta vez aplique **filtragem na ação derivativa**.
- Compare as *saídas de ambos os circuitos*, analisando como a filtragem afeta a resposta do sistema com ruído.