

José E. de A. Junior: 20170009356

Kallil de A. Bezerra: 20180154987

Rafael de M. M. Capuano: 20180010172

Victor K. C. Sousa: 20180155278

Controle PID de Sistemas Dinâmicos: Sistema de Primeira Ordem

Brasil

9 de outubro de 2020

José E. de A. Junior: 20170009356

Kallil de A. Bezerra: 20180154987

Rafael de M. M. Capuano: 20180010172

Victor K. C. Sousa: 20180155278

Controle PID de Sistemas Dinâmicos: Sistema de Primeira Ordem

Modelo canônico de Relatório Técnico e/ou Científico em conformidade com as normas ABNT apresentado à comunidade de usuários L^AT_EX.

Universidade Federal do Rio Grande do Norte – UFRN
Departamento de Engenharia da Computação e Automação – DCA

Brasil

9 de outubro de 2020

Resumo

Resumindo o início, meio e fim do trabalho

Palavras-chaves: Sistema de Primeira Ordem. PID. Sistemas Dinâmicos.

Lista de ilustrações

Figura 1.1 – Planta PID simples	8
Figura 2.1 – Controlador Proporcional	12
Figura 2.2 – Controlador P - Ganho 1	13
Figura 2.3 – Controlador P Ganho 1 - Sinal de entrada	13
Figura 2.4 – Controlador P - Ganho 5	14
Figura 2.5 – Controlador P - Ganho 10	14
Figura 2.6 – Controlador P Ganho 5 - Sinal de entrada	15
Figura 2.7 – Controlador P Ganho 10 - Sinal de entrada	15
Figura 2.8 – Controlador PID no Simulink	16
Figura 2.9 – Controlador PI - nível dos tanques - $K_p = 1$ e $K_i = 0,01$	17
Figura 2.10–Controlador PI - sinal de controle - $K_p = 1$ e $K_i = 0,01$	17
Figura 2.11–Controlador PI - ações de controle - $K_p = 1$ e $K_i = 0,01$	17
Figura 2.12–Controlador PID - nível dos tanques - $K_p = 1$, $K_i = 0,01$ e $K_d = 0,001$	18
Figura 2.13–Controlador PI - sinal de controle - $K_p = 1$, $K_i = 0,01$ e $K_d = 0,001$	18
Figura 2.14–Controlador PI - ações de controle - $K_p = 1$, $K_i = 0,01$ e $K_d = 0,001$	19
Figura 2.15–Controlador PID - nível dos tanques - $K_p = 2$, $K_i = 0,01$ e $K_d = 0,001$	19
Figura 2.16–Filtro derivativo	20
Figura 2.17–Filtro <i>anti windup</i>	20
Figura 2.18–Controlador P - Sistema de Segunda Ordem	21
Figura 2.19–Sinal de controle - Sistema de Segunda Ordem	21
Figura 2.20–Controlador PI - Sistema de Segunda Ordem	22
Figura 2.21–Sinal de controle PI - Sistema de Segunda Ordem	23
Figura 2.22–Controlador PID - Sistema de Segunda Ordem	24
Figura 2.23–Sinal de controle PID - Sistema de Segunda Ordem	24
Figura 2.24–Sinal de controle PID com filtro derivativo - Sistema de Segunda Ordem	25
Figura 2.25–Controlador PID em cascata - filtro derivativo	26
Figura 2.26–Controlador PID em cascata - filtro <i>anti-reset windup</i>	26
Figura 2.27–Controlador PID em cascata 2 - filtro <i>anti-reset windup</i>	27

Lista de tabelas

Tabela 2.1 – Valores do controlador PI	16
--	----

Sumário

	Introdução	6
1	EMBASAMENTO TEÓRICO	7
1.1	Controlador Proporcional (P)	7
1.2	Controlador Proporcional Integral (PI)	7
1.3	Controlador Proporcional Derivativo (PD)	8
1.4	Controlador Proporcional Integral Derivativo (PID)	8
1.5	Filtros	9
1.5.1	Filtro derivativo	9
1.5.2	Filtro <i>anti windup</i>	9
1.6	Sistemas de segunda ordem	9
1.6.1	Sistemas de segurança	10
1.7	Controle em cascata	11
2	DESENVOLVIMENTO E RESULTADOS	12
2.1	Implementação da ação proporcional - controlador P	12
2.1.1	Controlador P - ganho 1	13
2.1.2	Controlador P - ganhos 5 e 10	14
2.2	Expansão do controlador	16
2.2.1	Implementação da ação integrativa - controlador PI	16
2.2.2	Implementação da ação derivativa - controlador PD e PID	18
2.2.3	Implementação do filtro na ação derivativa	20
2.2.4	Implementação do filtro <i>anti-reset-windup</i>	20
2.3	Controlador de Segunda Ordem	21
2.3.1	Implementação do controlador proporcional em segunda ordem	21
2.3.2	Implementação do controlador proporcional integral em segunda ordem	22
2.3.3	Implementação do controlador PID em segunda ordem	24
2.3.4	Implementação do controlador PID em segunda ordem com filtro derivativo	25
2.4	Controle em cascata	25
2.4.1	Controlador PID com filtro derivativo	25
2.4.2	Controlador PID com filtro derivativo e <i>anti-reset windup</i>	26
3	CONCLUSÃO	28
	REFERÊNCIAS	29

Introdução

O controle automático tem desempenhado um papel vital no avanço da engenharia e da ciência. Além da sua importância em sistemas de veículos espaciais, sistemas robóticos, e semelhantes, o controle automático tornou-se uma importante parte da fabricação moderna e dos processos industriais. É possível citar o controle numérico de ferramentas e máquinas nas indústrias de manufatura, no projeto de sistemas de piloto automático em operações aeroespaciais e no projeto de carros e caminhões na indústria automobilística. O controle também é essencial no controle de pressão, temperatura, umidade e viscosidade nos processos industriais (OGATA, 2001).

Os avanços na teoria e na prática do controle fornecem meios para atingir o desempenho desejado dos sistemas dinâmicos, melhorando a produtividade e aliviando o trabalho penoso de algumas atividades. Portanto, os engenheiros e cientistas precisam ter uma boa compreensão deste campo para poder aplicar e entender o que está sendo aplicado na prática.

1 Embasamento teórico

O controlador PID esteve em uso por mais de um século em várias formas e aplicações. Já foi popular como um dispositivo puramente mecânico, também como um dispositivo pneumático e até eletrônico. Atualmente o PID é implementado em sistemas embarcados, e os microprocessadores são essenciais nessa tarefa ([WESCOTT, 2020](#))

As três letras que compõem o PID vem de Proporcional, Integral e Derivativo, e cada um desses elementos tem uma tarefa diferente, portanto, causam diferentes efeitos na funcionalidade de um sistema. Num típico controle PID esses elementos são orientados por uma combinação de comandos do sistema e de respostas do sinal que está sendo controlado.

1.1 Controlador Proporcional (P)

O controle Proporcional é o tipo de resposta mais simples, ele é composto pelo erro sendo multiplicado por uma constante e adicionado, ou alimentado, ao *drive*.

$$U(s) = K_p E(s) \quad (1.1)$$

Dentre as características do controlador P podemos citar ([ARAÚJO, 2020](#)):

- O controlador proporcional é um amplificador com ganho ajustável (K);
- O aumento do ganho K diminui o erro do regime;
- Em geral, o aumento de K torna o sistema mais oscilatório, podendo gerar instabilidade;
- Melhora o regime e piora o transitório, sendo bastante limitado.

1.2 Controlador Proporcional Integral (PI)

O controle Integral é usado para adicionar uma precisão de longo termo ao *loop* do controle, normalmente é usado com controle Proporcional.

$$U(s) = \frac{(K_p s + K_i)}{s} E(s) \quad (1.2)$$

Dentre as características do controlador PI podemos citar ([ARAÚJO, 2020](#)):

- Zera o erro de regime, pois aumenta o tipo do sistema em 1 unidade;

- É utilizado quando tempos resposta transitória aceitável e resposta em regime insatisfatória;
- Adiciona um polo em $p = 0$ e um zero em $z = \frac{-K_i}{K_p} = \frac{-1}{\tau_i}$;
- Como aumenta a ordem do sistema, tempos a possibilidade de instabilidade do sistema original. Pode degradar o desempenho do controlador em malha fechada.

1.3 Controlador Proporcional Derivativo (PD)

É conhecido que o controle proporcional trabalha com o comportamento presente da planta, e o integral trabalha com o passado do sistema. O controlador derivativo tenta prever como será o comportamento do sistema, tentando adiantar a estabilidade. O componente diferencial é, resumidamente, a última posição conhecida menos o valor atual da posição.

$$U(s) = (K_p + K_d s)E(s) \quad (1.3)$$

Dentre as características do controlador PD podemos citar ([ARAÚJO, 2020](#)):

- Leva em conta a taxa de variação do erro;
- É utilizado quando temos resposta em regime aceitável e resposta transitória insatisfatória;
- Adiciona um zero em $z = \frac{-K_p}{K_d} = \frac{-1}{\tau_d}$;
- Introduce um efeito de antecipação no sistema, fazendo com que o mesmo reaja não somente à magnitude do sinal de erro, como também à sua tendência para o instante futuro, iniciando, assim, uma ação corretiva mais cedo.

1.4 Controlador Proporcional Integral Derivativo (PID)

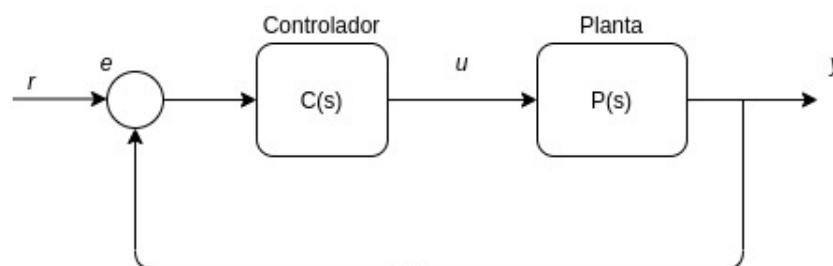


Figura 1.1 – Planta PID simples

De forma resumida, o controlador PID, trabalhando numa malha fechada, tem a estrutura mostrada na imagem acima. A variável e representa o erro, que é a diferença entre

a saída desejada r e a saída real y . Esse sinal de erro é alimentado para o controlador PID, que calcula tanto a derivada como a integral desse sinal de erro em relação ao tempo. O sinal do controlador, u , para a planta é igual ao ganho proporcional K_p vezes a magnitude do erro mais a integral do ganho K_i vezes a integral do erro mais a derivada do ganho K_d vezes a derivada do erro.

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de}{dt} \quad (1.4)$$

Esse sinal de controle, u , é passado para a planta e a nova saída y é obtida. Essa nova saída é enviada de volta e comparada à referência para que se encontre o novo sinal de erro. O controlador recebe esse novo sinal e atualizada a entrada do controlador. Esse processo continua enquanto o controlador estiver funcionando.

1.5 Filtros

Os filtros têm por objetivo tornar o controle implementado, diminuindo o sobressinal e o tempo para se alcançar o *setpoint*.

1.5.1 Filtro derivativo

O filtro derivativo age no regime transitório. Esse tipo específico de filtro tenta prever alguns comportamentos, eliminando ruídos de alta frequência e suavizando o efeito do ruído no controle. No caso real a queda da água, por exemplo, poderia ser considerada como um ruído, porque essa queda causaria um distúrbio no tanque. Na simulação é um pouco mais difícil de notar esse tipo de comportamento, então, adiantando a parte prática, encontramos um pouco de dificuldade de ver o efeito dele no sistema.

1.5.2 Filtro *anti windup*

Sistemas de controle podem sofrer com resposta transitória lenta e oscilatória se houver, como entrada, um sinal saturado, pois a malha de realimentação é desfeita quando esse valor não é considerado. Isso faz com que a saída demore muito para atingir o *setpoint* e apresente oscilações frequentes, esses dois comportamentos são indesejáveis numa planta industrial. Então, para atacar esse problema existe o filtro *anti windup*, que faz com que o controlador saia da região da saturação e volte para o comportamento esperado.

1.6 Sistemas de segunda ordem

Até aqui foram discutidos os conceitos necessários para a realização da primeira parte do experimento, para a segunda parte serão necessários alguns novos conceitos relacionados

aos sistemas de segunda ordem.

Os sistemas de segunda ordem são aqueles cujo modelo pode ser escrito por uma equação diferencial de segunda ordem, ou seja, os que possuem dois polos ([MADEIRA, 2016](#)).

O comportamento dos sistemas de segunda ordem é bastante sensível à mudança nos parâmetros, então é importante fazer uma classificação dos perfis existentes para esses sistemas, tentando entender melhor os fenômenos que ocorrem em cada caso. Durante as simulações foi notado que apesar de haver bastante matemática por trás dos modelos, muita coisa pode ser entendida de forma mais fácil de forma empírica, alterando os parâmetros e analisando a mudança nos gráficos.

Existem três tipos principais de sistemas de segunda ordem, são eles:

1. Superamortecido: um polo na origem, proveniente da entrada degrau unitário e dois polos reais;
2. Subamortecido: um polo na origem proveniente da entrada degrau unitário e dois polos complexos, vindos do sistema;
3. Criticamente amortecido: um polo na origem proveniente da entrada degrau unitário e dois polos reais iguais.

1.6.1 Sistemas de segurança

No segundo experimento será necessário implementar um sistema de segurança que impeça a água de transbordar, mas a bomba também não pode trabalhar sem água, numa situação real ela poderia queimar. Então será feito um sistema de segurança que não permita nem que haja um transbordamento nem uma quebra do motor.

Esse tipo de segurança é definido como sendo um conjunto de passos, na lógica de automação e controle, que devem existir para garantir a segurança de um equipamento, pessoa ou processo ([ARAÚJO, 2020](#)). Nesses casos as travas lógicas, que são implementadas em *software* e as travas físicas, que são *hardware*, são usadas para garantir a segurança das pessoas e a continuidade da produção.

O sistema vai monitorar certas condições da planta, tentando mantê-la dentro do limite operacional estabelecido, e quando condições de risco, ou que estiverem fora dos parâmetros criados, forem detectadas ele deve acionar alarmes, ou também pode atuar nos erros.

Na simulação as seguintes regras devem ser seguidas e implementadas no sistema de segurança:

- Não deve ser enviado para o sistema uma tensão fora dos limites de $\pm 4V$;
- A bomba não pode trabalhar sem que haja líquido a ser bombeado;

- Nenhum dos dois tanques pode transbordar.

1.7 Controle em cascata

O controle em cascata é implementado quando a malha de controle simples não responde satisfatoriamente, principalmente em processos que possuem uma perturbação contínua em torno da variável que se deseja manipular, ou seja, o nível do tanque 2 com o constante recebimento de água a partir do tanque 1 pode se encaixar nessa situação. O controle em cascata consiste de um controlador primário que regula um controlador secundário, melhorando a velocidade de resposta e reduzindo os distúrbios causados pela malha secundária

No arranjo proposto pelo roteiro pode-se manipular a tensão enviada para a bomba e obter o nível desejado no tanque 1, que influencia na vazão de saída afetando a vazão de entrada do tanque 2. O objetivo é controlar o nível do segundo tanque a partir do primeiro, então com uma malha externa de controle é preciso captar o erro entre a referência desejada e o nível do tanque 2 e fornecer um valor para o tanque 1, alterando a tensão do motor e aumentando, ou diminuindo, o nível do segundo tanque.

2 Desenvolvimento e resultados

O experimento consiste dos seguintes pontos:

1. Implementar a ação proporcional, formando controladores P, e realizar testes com diferentes valores de ganho;
2. Realizando testes em cada etapa, faça o proposto:
 - a) Implemente a ação integrativa, formando controladores PI;
 - b) Implemente a ação derivativa, permitindo agora formar controladores PD e PID;
 - c) Implemente o filtro na ação derivativa;
 - d) Implemente o filtro *anti-reset-windup*.
3. Descrever a diferença no comportamento do sistema com cada uma das leis de controle implementada. Levando em consideração as entradas degrau e senoide;
4. Para cada lei de controle, verificar e descrever o comportamento do sistema para diferentes valores dos ganhos.

2.1 Implementação da ação proporcional - controlador P

Como foi dito anteriormente, esse é o controlador mais simples, pois o sinal de saída é proporcional ao erro. Para implementar esse tipo de controlador basta colocar um ganho entre o comparador e o bloco de saturação.

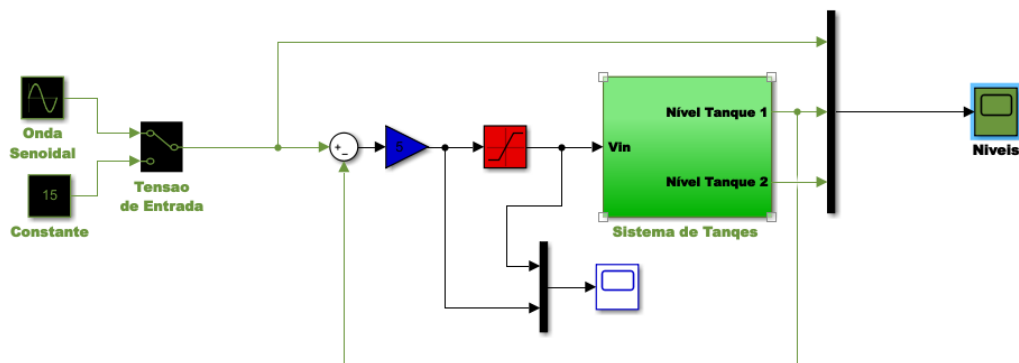


Figura 2.1 – Controlador Proporcional

Com isso feito, a simulação apresenta o seguinte comportamento de acordo com os ganhos.

2.1.1 Controlador P - ganho 1

O controlador P é o mais simples, ele apenas vai tentar compensar o erro de forma proporcional. Começamos com ganho unitário, ou seja, praticamente não haverá correção do erro.

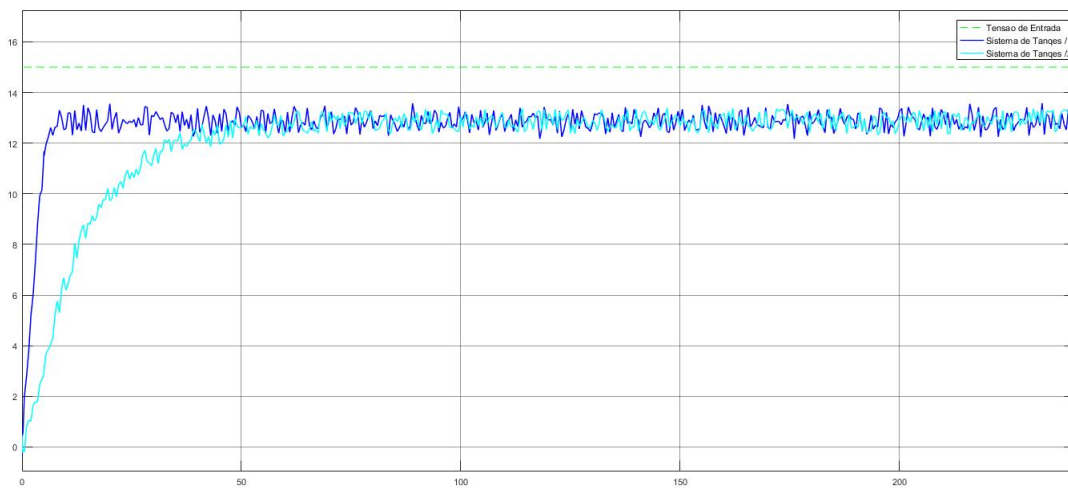


Figura 2.2 – Controlador P - Ganho 1

Na imagem acima é possível ver que o ganho unitário não alcança o valor desejado, então fica abaixo do que seria ideal, 15. Não há muito o que discutir nesse caso, mas nos próximos ganhos, de 5 e 10, veremos diferenças.

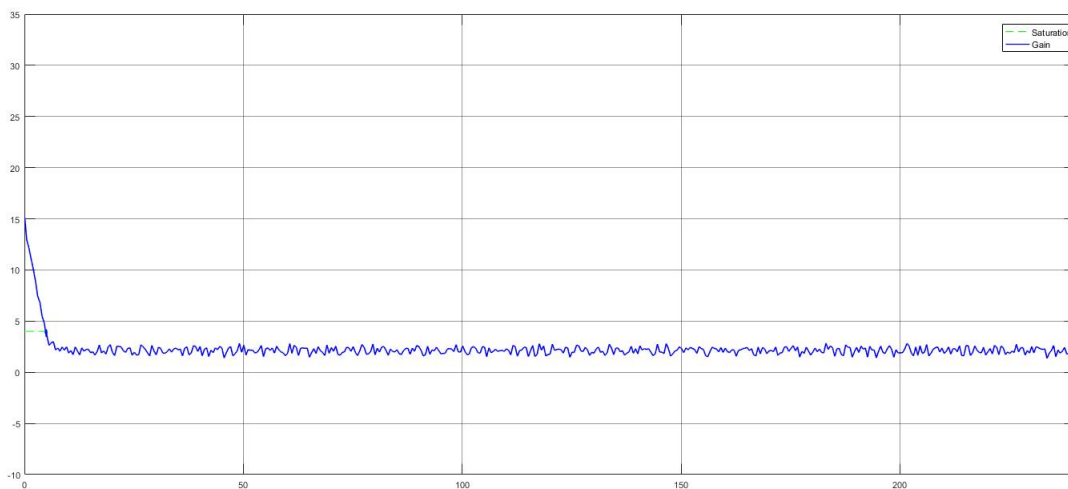


Figura 2.3 – Controlador P Ganho 1 - Sinal de entrada

2.1.2 Controlador P - ganhos 5 e 10

No caso de ganhos maiores, 5 e 10 no exemplo apresentado, o controlador vai aplicar esses valores tentando corrigir o erro encontrado, se o ganho for muito alto, o controlador vai aplicar "mais vezes" tentando compensar a maior distância em relação ao valor ideal, de novo, 15. Esse comportamento pode ser visto no nível, como mostrado a seguir.

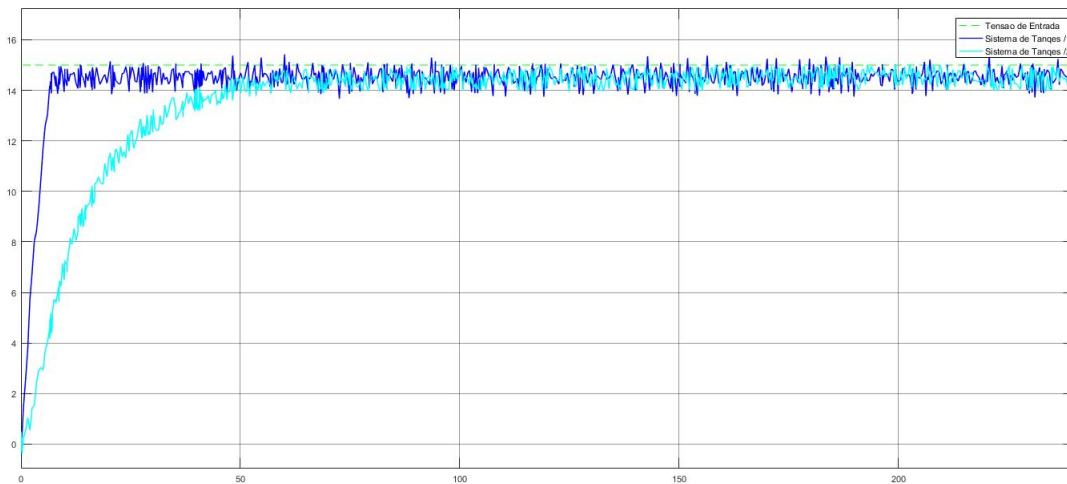


Figura 2.4 – Controlador P - Ganho 5

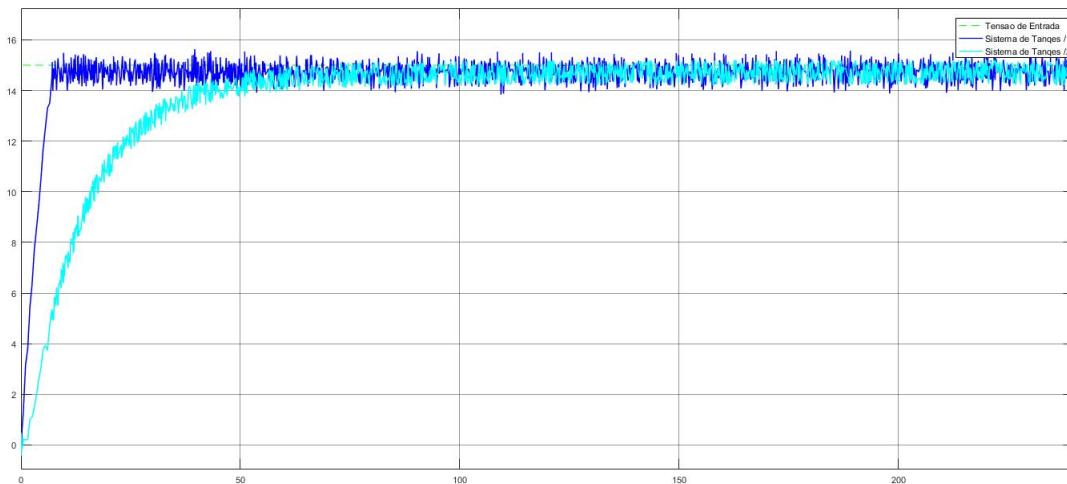


Figura 2.5 – Controlador P - Ganho 10

E também pode ser visto no sinal do controlador, como pode ser visto abaixo.

O comportamento de compensação do ganho fica muito nítido quando comparamos as figuras 2.6 e 2.7, em que o Controlador com Ganho 10 aplica o sinal de forma mais frequente e tem uma amplitude maior também, justamente porque o ganho é muito alto e a cada vez que ele aplica esse ganho se distancia bastante do valor que precisa atingir, então aplica um sinal novamente e, de novo, se distancia do objetivo. Esse comportamento se repete constantemente e gera as imagens que podemos analisar.

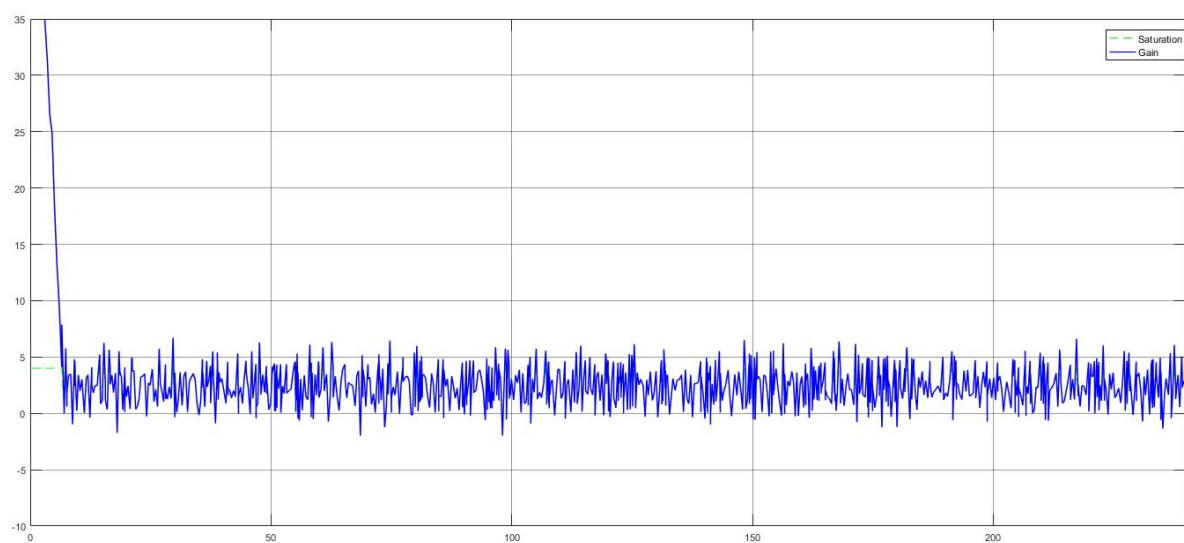


Figura 2.6 – Controlador P Ganho 5 - Sinal de entrada

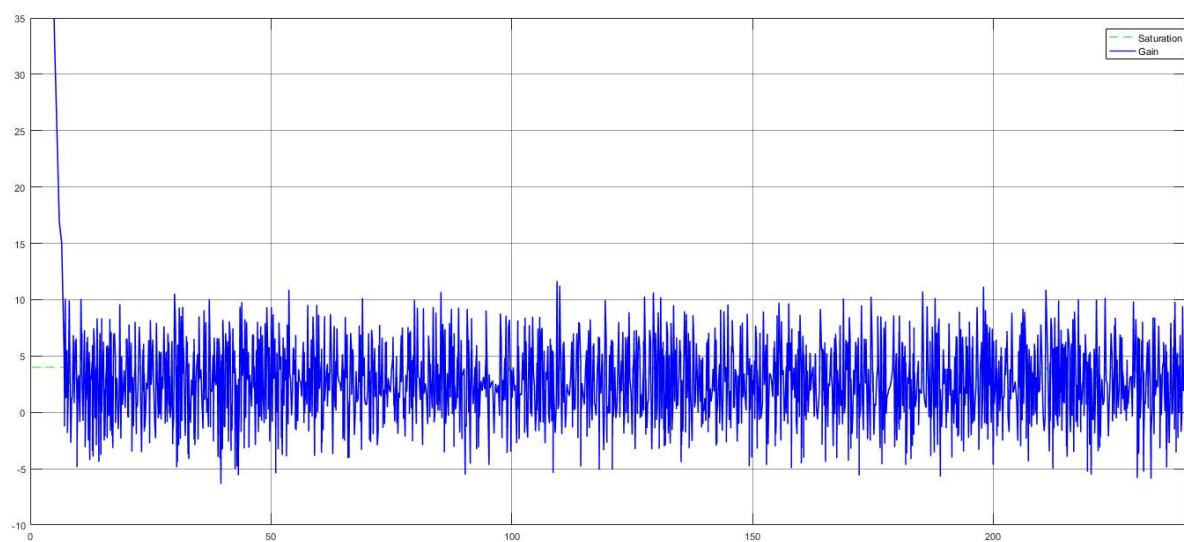


Figura 2.7 – Controlador P Ganho 10 - Sinal de entrada

2.2 Expansão do controlador

Nessa parte, ainda nos sistemas de primeira ordem, vamos expandir o controlador, adicionando os componentes Integrador e Derivativo. Para realizar esses experimentos, construímos uma simulação já com os três componentes do controlador PID, e quando queríamos apenas um PD, por exemplo, deixamos o K_i igual a 0, então não havia influência do Integrador no sistema.

A simulação usada foi construída usando como base o arquivo disponibilizado no SIGAA.

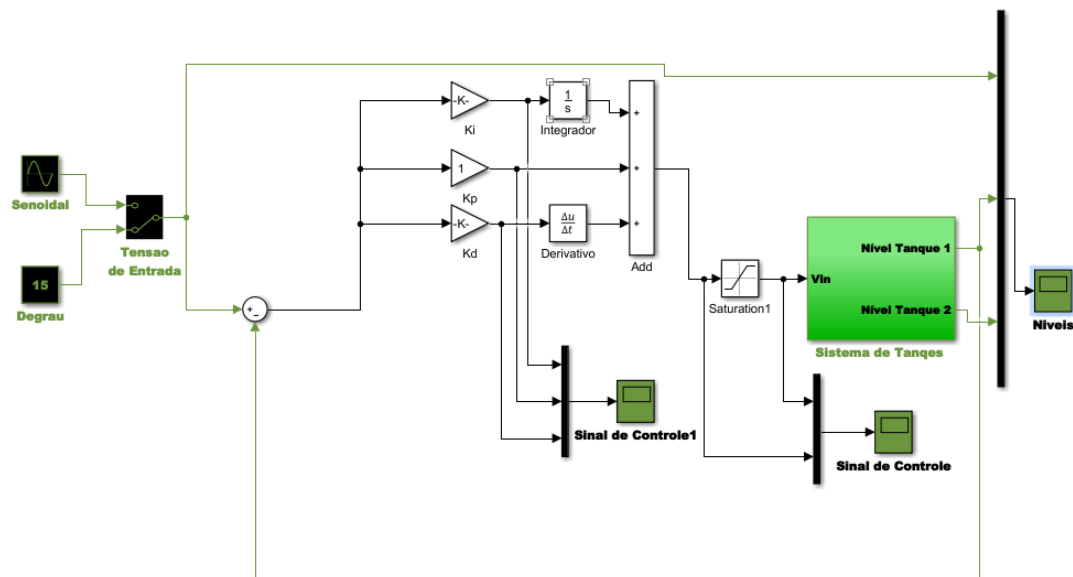


Figura 2.8 – Controlador PID no Simulink

2.2.1 Implementação da ação integrativa - controlador PI

O termo integral é proporcional à magnitude do erro e à duração do erro. De forma mais detalhada, a integral no PID é a soma dos erros ao longo do tempo, esse termo vai tentar corrigir os erros deslocando a curva. Ele vai acelerar a aproximação ao *setpoint*, e por isso pode causar um *overshoot*. Usamos os valores da tabela 2.1 no experimento.

K_p	K_i	K_d
1,0	0,01	0
1,0	0,02	0
2,0	0,01	0

Tabela 2.1 – Valores do controlador PI

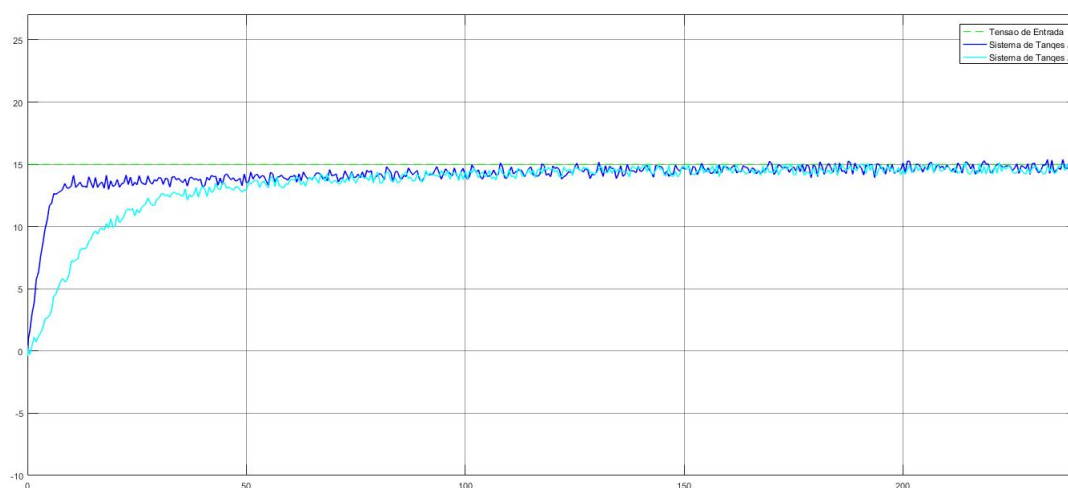


Figura 2.9 – Controlador PI - nível dos tanques - $K_p = 1$ e $K_i = 0,01$

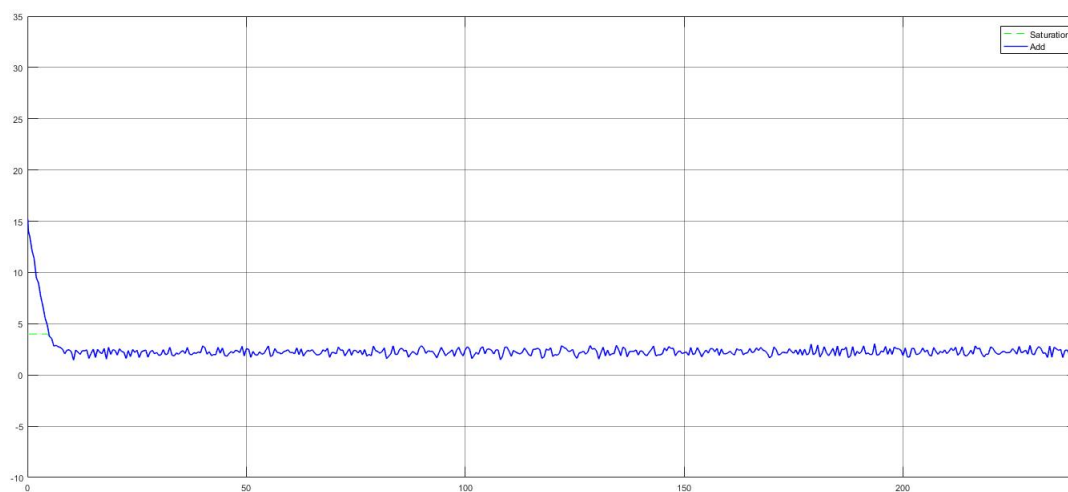


Figura 2.10 – Controlador PI - sinal de controle - $K_p = 1$ e $K_i = 0,01$

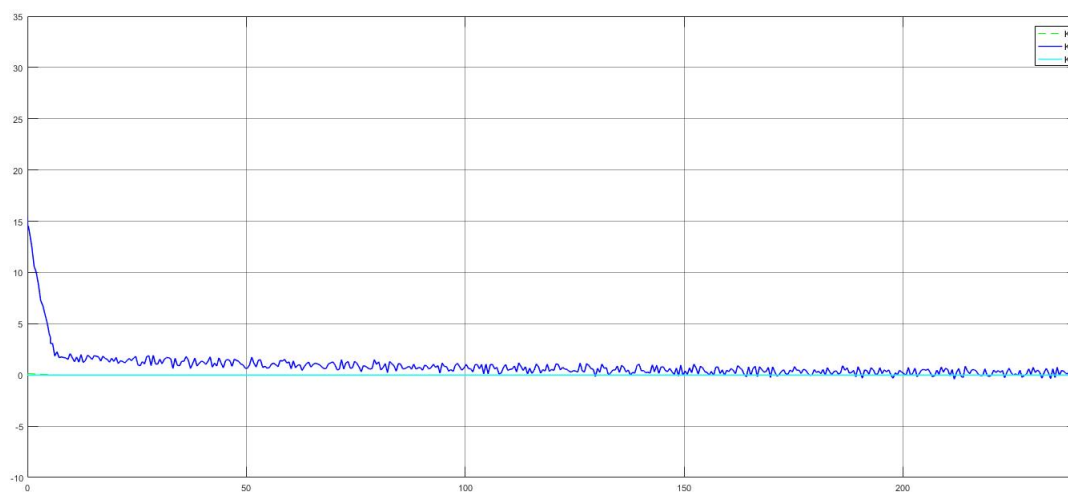


Figura 2.11 – Controlador PI - ações de controle - $K_p = 1$ e $K_i = 0,01$

Nas imagens 2.9, 2.10 e 2.11 é possível notar que o controlador demorou um pouco mais que o tipo P somente para alcançar o *setpoint*, mas uma vez alcançado, o comportamento é bem definido e estável, no sentido de não se distanciar mais do objetivo. É importante testar um controlador com o K_p alto para ver o impacto que o K_i pode causar.

2.2.2 Implementação da ação derivativa - controlador PD e PID

A ação derivativa tenta prever o comportamento do sistema diminuindo o tempo em que se alcança a estabilidade. E esse foi o comportamento notado quando aplicamos a ação derivativa, uma diminuição de aproximadamente 50% para se alcançar a estabilidade, e uma diminuição do *overshoot*, em alguns casos o *overshoot* nem acontecia.

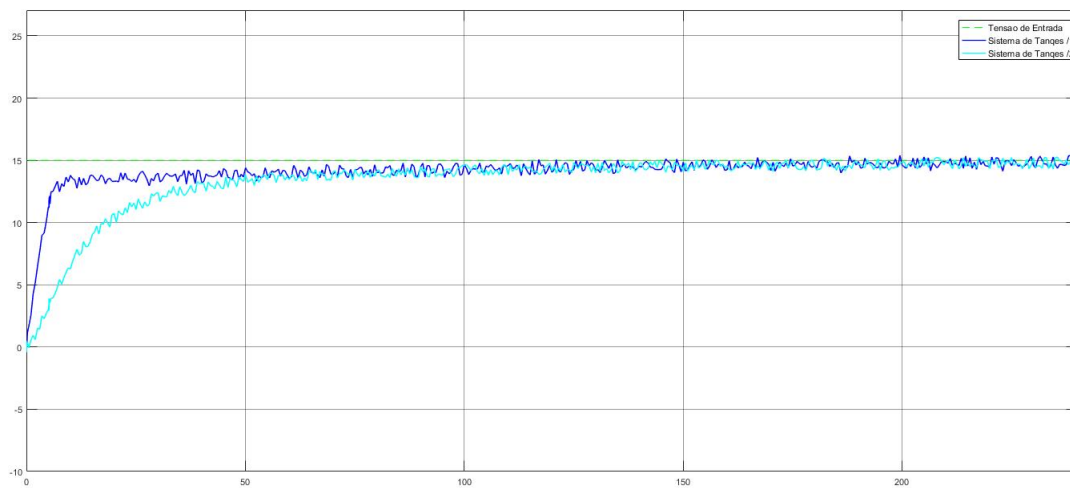


Figura 2.12 – Controlador PID - nível dos tanques - $K_p = 1$, $K_i = 0,01$ e $K_d = 0,001$

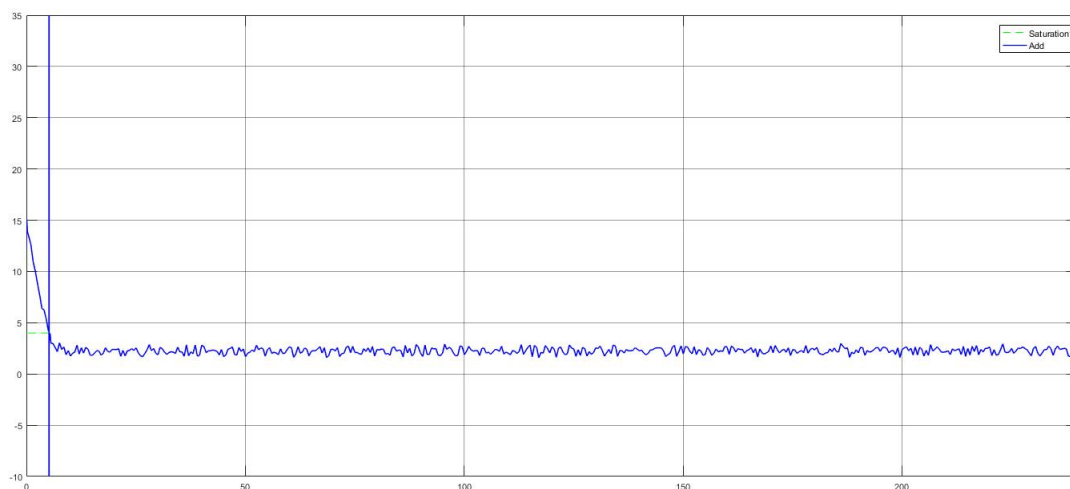


Figura 2.13 – Controlador PI - sinal de controle - $K_p = 1$, $K_i = 0,01$ e $K_d = 0,001$

O *overshoot* pode ser visto na figura 2.15. Então o conjunto das três ações possíveis trabalha bem, diminuindo o *overshoot* e diminuindo o tempo para se alcançar a estabilidade.

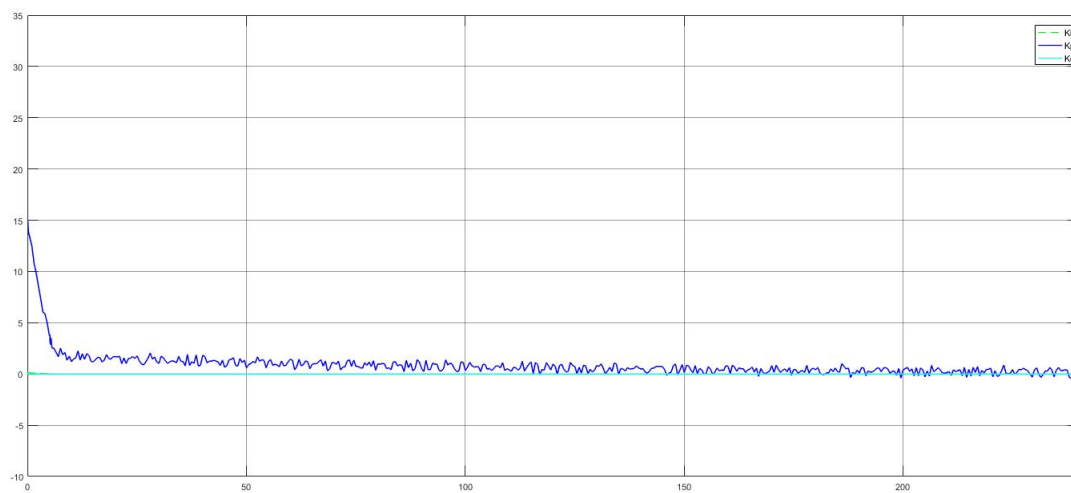


Figura 2.14 – Controlador PI - ações de controle - $K_p = 1$, $K_i = 0,01$ e $K_d = 0,001$

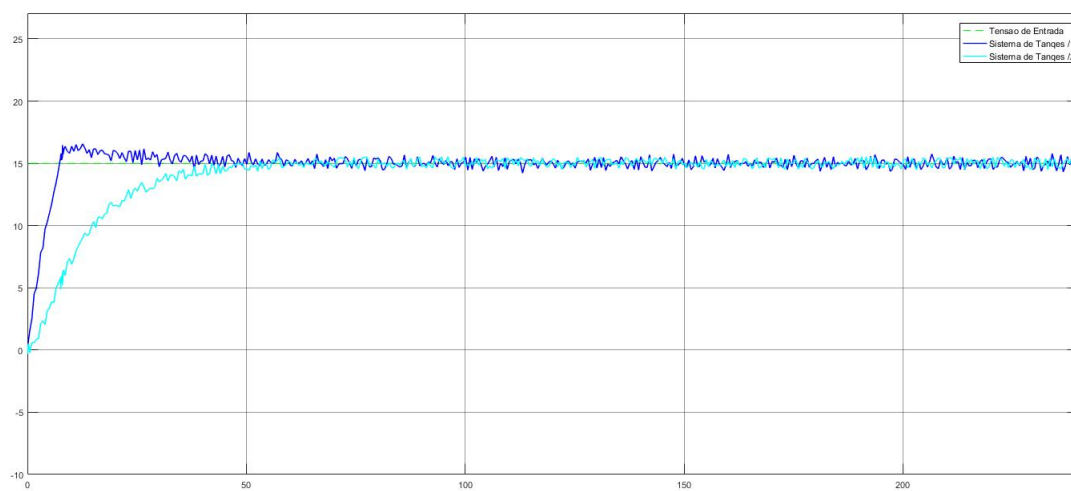


Figura 2.15 – Controlador PID - nível dos tanques - $K_p = 2$, $K_i = 0,01$ e $K_d = 0,001$

2.2.3 Implementação do filtro na ação derivativa

No filtro de ação derivativa obtivemos resultados que não foram muito satisfatórios, mesmo alterando os valores dos coeficientes o comportamento era praticamente o mesmo. Apesar de ter apresentado um menor tempo para alcançar a acomodação, a diferença não foi grande, o melhor ponto que pode ser citado aqui é a diminuição do sobressinal.

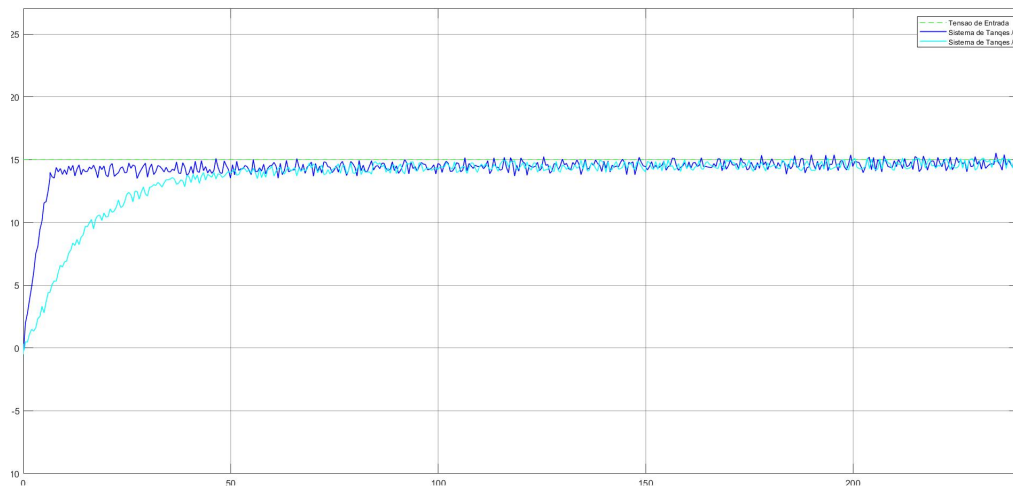


Figura 2.16 – Filtro derivativo

2.2.4 Implementação do filtro *anti-reset-windup*

O filtro *anti windup* deveria apresentar um sinal melhor que o controlador PID simples, em resumo, ele deveria apresentar menor sobressinal e menor tempo para atingir a estabilidade, porém, obtivemos um comportamento semelhante ao filtro de ação derivativa. Apesar de realmente apresentar uma melhora, é uma mudança muito pequena em relação ao controlador PID convencional, talvez numa planta industrial real a dificuldade de se implementar um filtro desses e o aumento na complexidade da manutenção não compensem o ganho marginal do sinal.

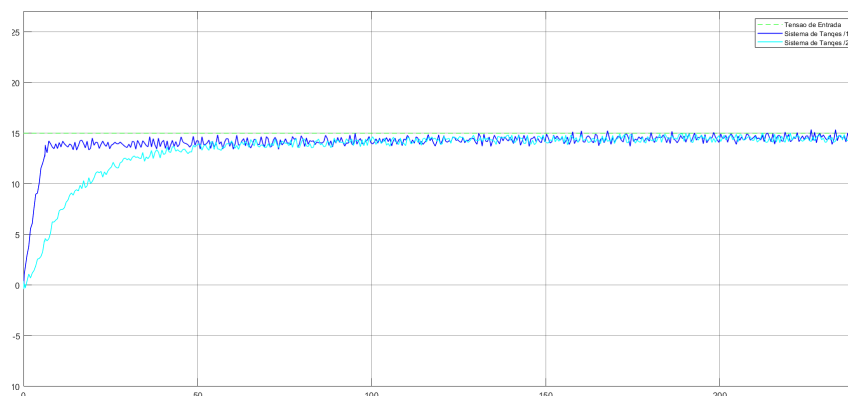


Figura 2.17 – Filtro *anti windup*

2.3 Controlador de Segunda Ordem

INSERIR TEXTO AQUI

2.3.1 Implementação do controlador proporcional em segunda ordem

Os testes começaram com a implementação do controlador proporcional, testamos com o $K_p = 1,0$, e obtemos os seguintes resultados:

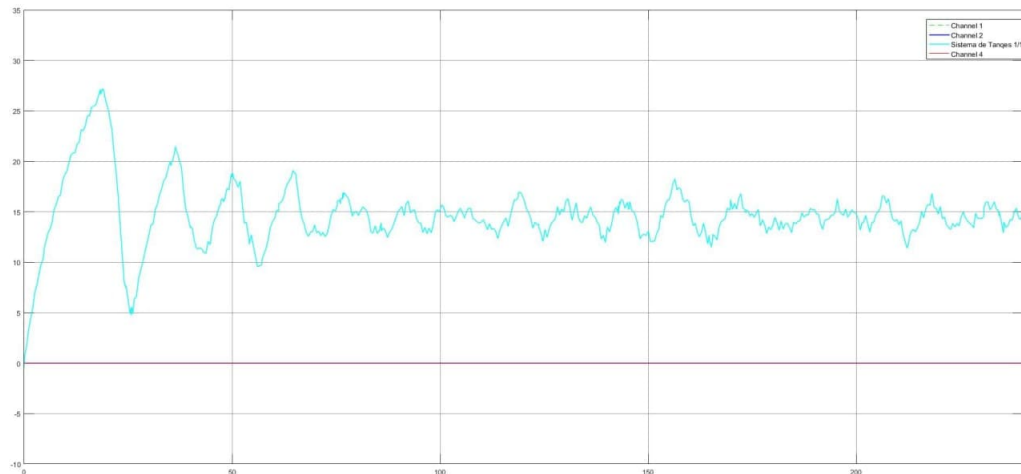


Figura 2.18 – Controlador P - Sistema de Segunda Ordem

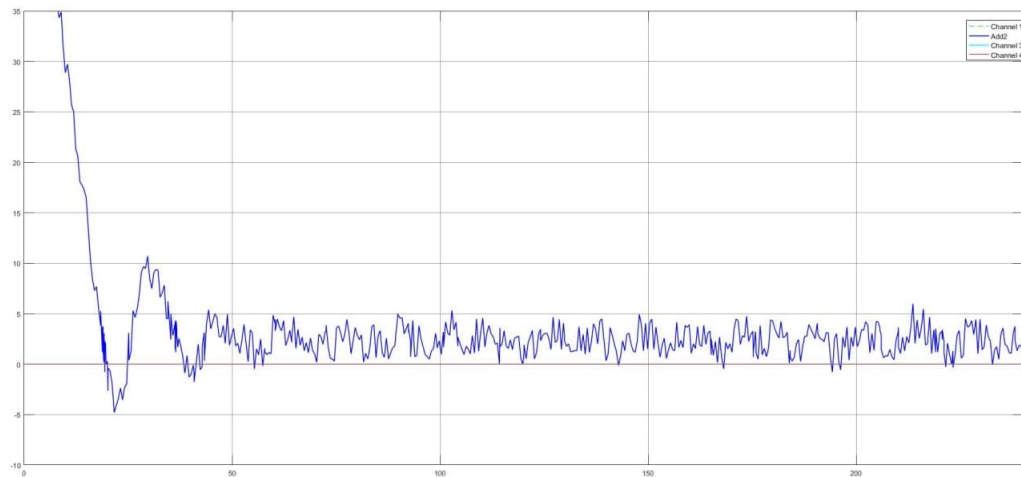


Figura 2.19 – Sinal de controle - Sistema de Segunda Ordem

O comportamento, no geral, é o esperado para essa situação porque o aumento do K_p torna o sistema bastante oscilatório e o erro de regime fica menor. O sinal de entrada começa em 35V e fica saturando por aproximadamente 25s, depois se torna bastante oscilatório. Após isso usamos um $K_p = 5,8$. O sistema teve um *overshoot* alto, porém houve menos oscilação e o erro de regime aumentou.

Como foi mostrado, não é recomendado controlar um sistema por meio de um controlador proporcional simples, porque com o aumento do K_p o erro de regime é reduzido ao custo de um sistema com mais oscilação. Comparando isso com uma possível aplicação do mundo real teríamos um sistema que não consegue manter uma boa reprodutibilidade de peças, por exemplo, elas estariam sempre próximo ao desejado, mas apresentariam diferenças entre si.

2.3.2 Implementação do controlador proporcional integral em segunda ordem

O controlador PI, como já foi dito, é a soma da ação proporcional com a integral, então com essa união a resposta transitória e o erro de regime melhoram, isso é causado pela ação proporcional e integral, respectivamente. Os testes usaram valores de $K_p = 3,0$ e $K_i = 0,015$.

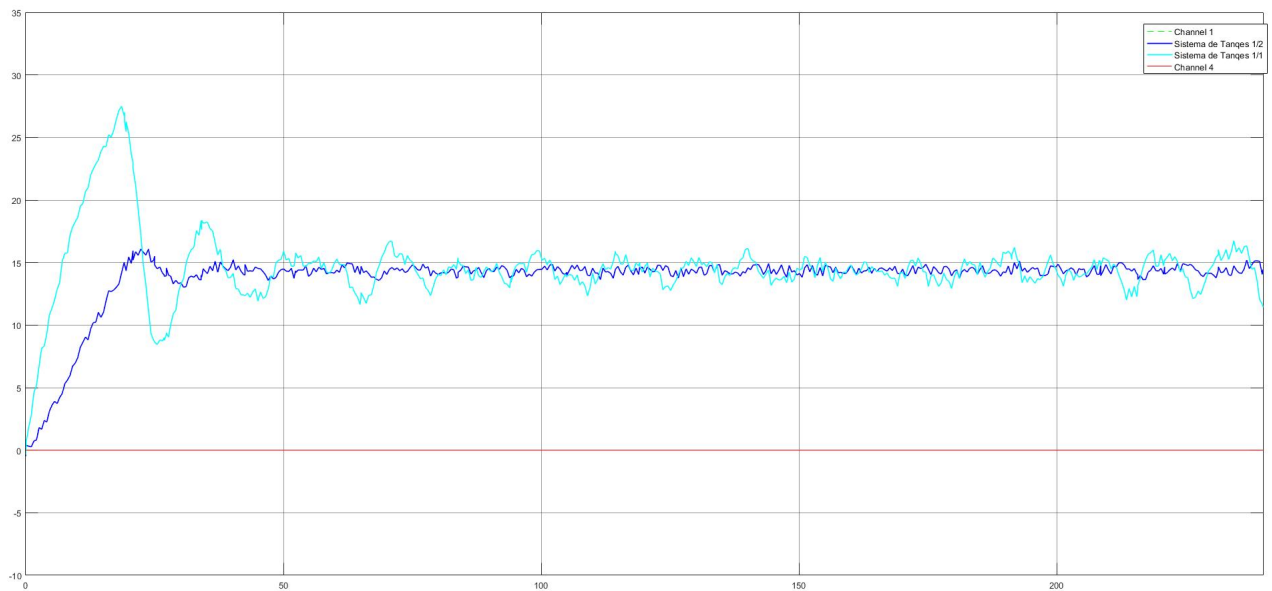


Figura 2.20 – Controlador PI - Sistema de Segunda Ordem

Os valores resultaram num *overshoot* alto, no tanque 1, mas no tanque 2 foi baixo. Notamos que o sinal oscila bastante, mas esse comportamento tem se repetido ao longo dos experimentos. Apesar desses problemas, o sistema alcança a estabilidade antes dos 25 segundos.

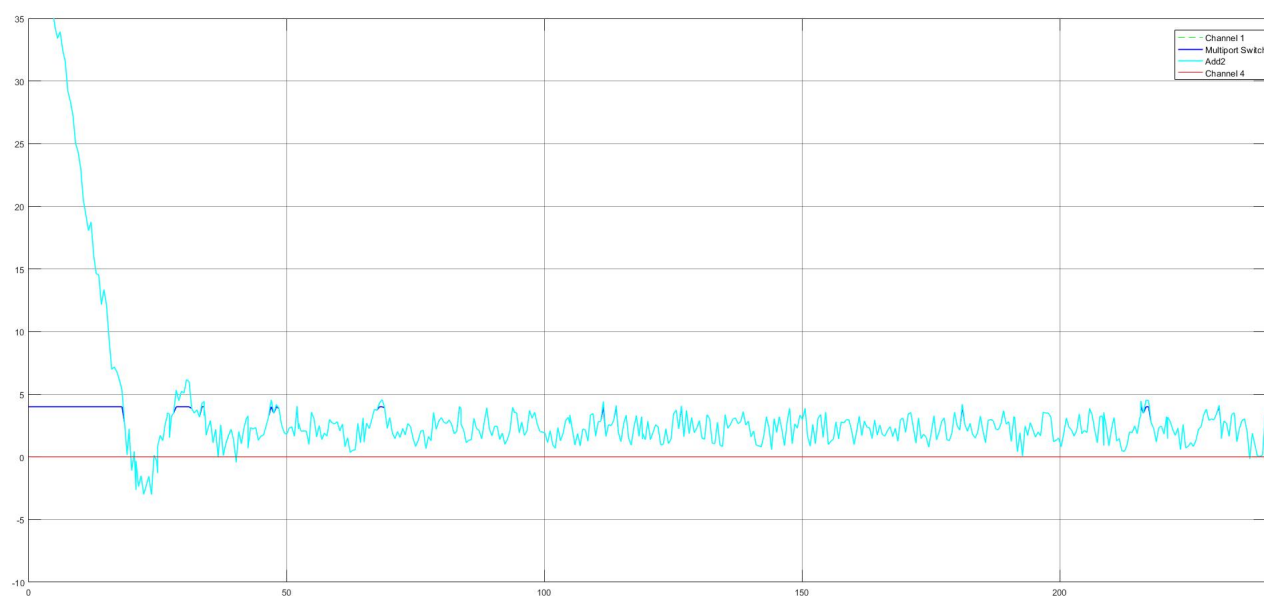


Figura 2.21 – Sinal de controle PI - Sistema de Segunda Ordem

2.3.3 Implementação do controlador PID em segunda ordem

O controlador PID reúne todas as ações vistas (proporcional, integral e derivativa), elas atuam no regime transitório e permanente. Utilizando os valores $K_p = 7,5$, $K_i = 0,015$ e $K_d = 0,001$, diminuimos o *overshoot*, mas não reduzimos a oscilação. O gráfico da saturação apresenta dois picos que chamam a atenção e não conseguimos consertar.

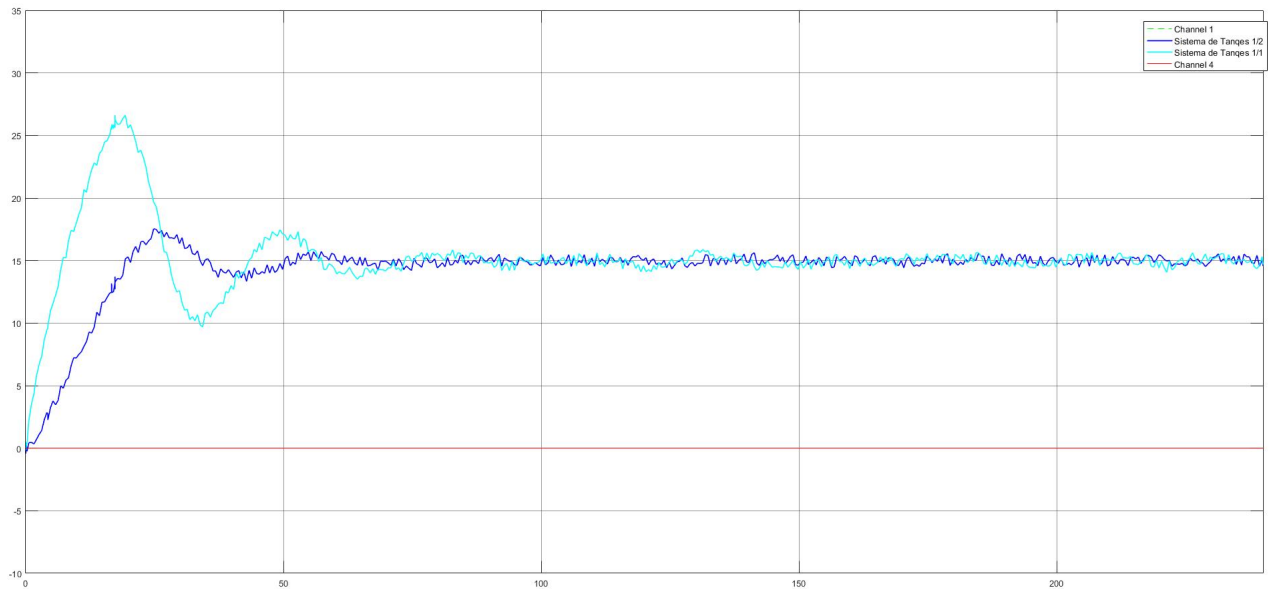


Figura 2.22 – Controlador PID - Sistema de Segunda Ordem

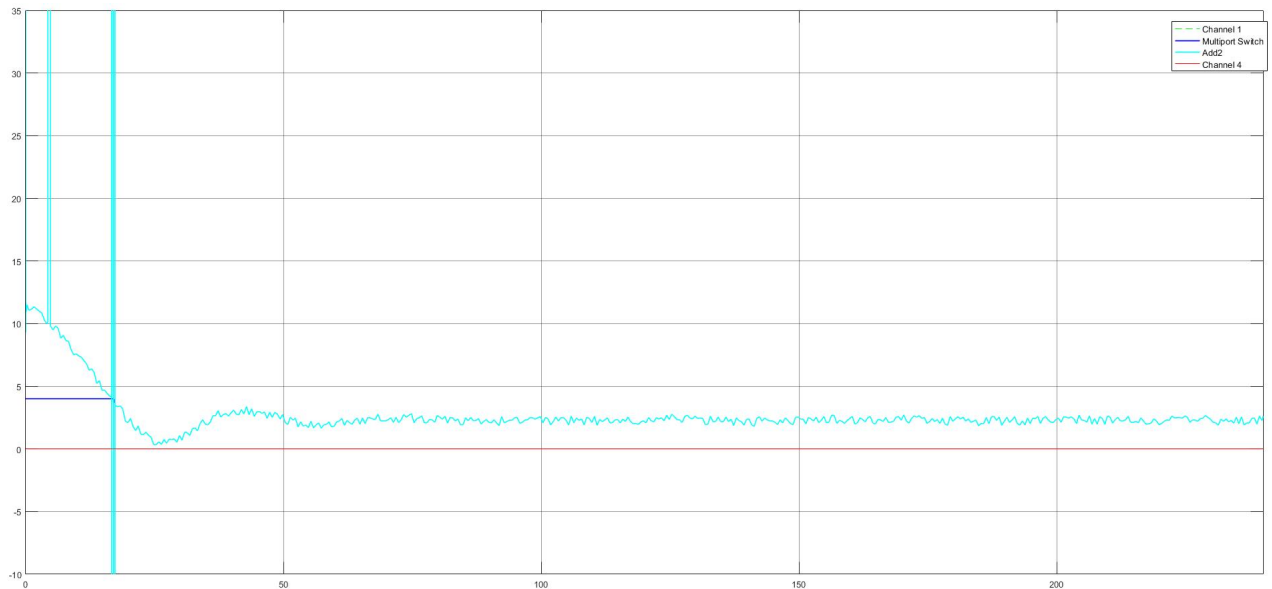


Figura 2.23 – Sinal de controle PID - Sistema de Segunda Ordem

Com esses valores o erro de regime não foi zerado, mas converge para o *setpoint*.

2.3.4 Implementação do controlador PID em segunda ordem com filtro derivativo

A combinação de controlador PID com o filtro derivativo foi a que rendeu os melhores resultados, diferentemente desse conjunto aplicado ao sistema de primeira ordem, o sistema de segunda ordem se comportou muito melhor, diminuindo o erro de regime e o sobressinal.

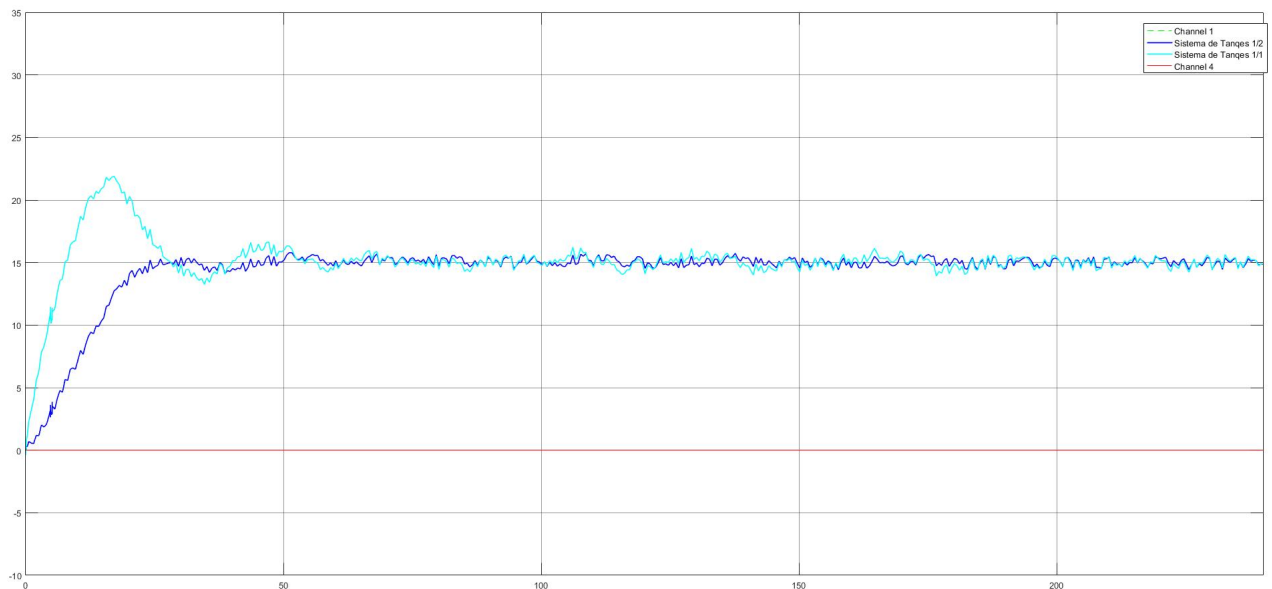


Figura 2.24 – Sinal de controle PID com filtro derivativo - Sistema de Segunda Ordem

No filtro *anti-reset windup* usamos 0.01, já para as ações usamos $K_p = 2,5$, $K_i = 0,025$ e $K_d = 0,01$, curiosamente o valor usado no filtro e usado no K_d é o mesmo, e dessa forma conseguimos o gráfico mostrado, mas não entendemos essa relação em específico. Foi notado, de forma empírica, que quando eles eram diferentes haviam mudanças mais significativas no gráfico.

2.4 Controle em cascata

Na última parte do segundo experimento usamos dois controladores, sempre com o objetivo de melhorar o desempenho do sistema.

2.4.1 Controlador PID com filtro derivativo

O primeiro teste com o controlador em cascata foi usando um filtro derivativo nos dois controladores, nesse experimento foi utilizado um $K_{pe} = 2,5$. Nessa configuração obtemos um controlador que não era muito preciso em relação ao segundo tanque por causa do sobressinal, no primeiro reservatório o controle foi satisfatório, porque não havia *overshoot* e o tempo de acomodação foi semelhante ao do segundo tanque.

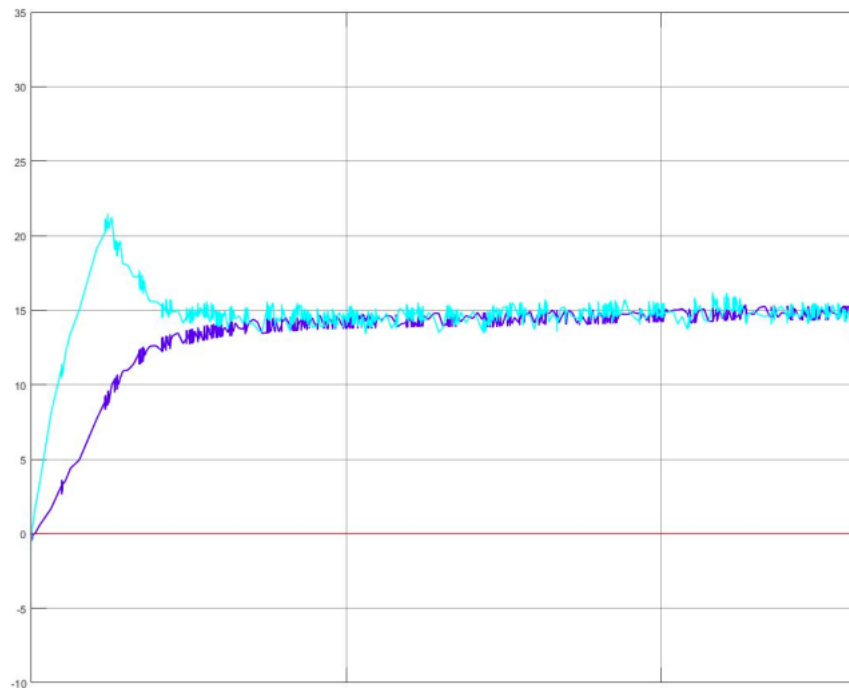


Figura 2.25 – Controlador PID em cascata - filtro derivativo

2.4.2 Controlador PID com filtro derivativo e *anti-reset windup*

Nessa configuração começamos com $K_p = 2$, $K_i = 0,06$ e $K_d = 0.003$ para o mestre, e $K_p = 1$, $K_i = 0,03$ e $K_d = 0.001$ para o escravo. Esses valores resultaram em uma anulação do *overshoot* e uma convergência do sistema ao nível desejado, apesar disso, o sinal da malha mestre apresenta bastante ruído. Realizamos mais alguns testes, para tentar melhorar o comportamento apresentado pelo gráfico 2.26, e assim obtivemos o gráfico 2.27.

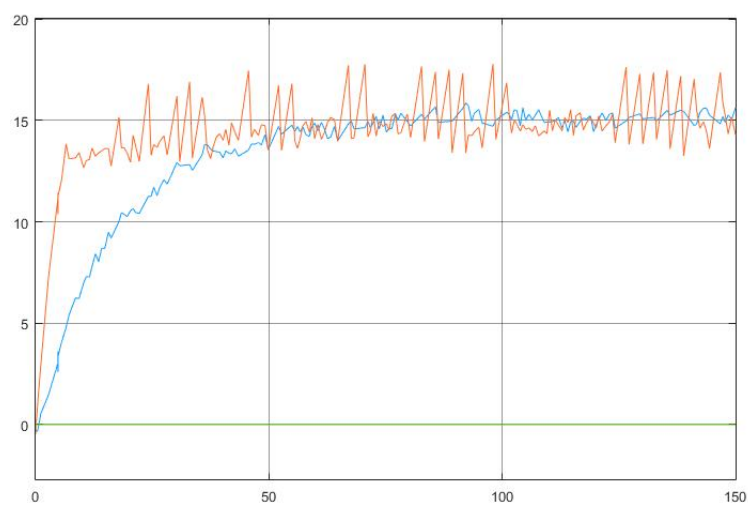
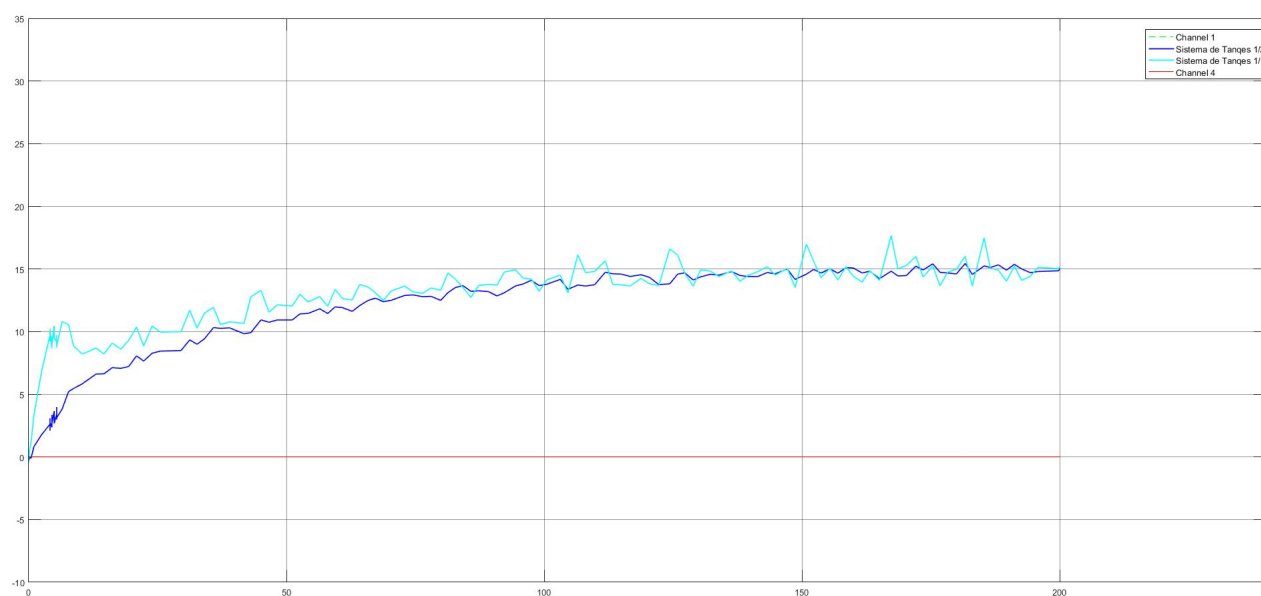


Figura 2.26 – Controlador PID em cascata - filtro *anti-reset windup*

Figura 2.27 – Controlador PID em cascata 2 - filtro *anti-reset windup*

3 Conclusão

As simulações permitiram que fosse melhor entendido o comportamento dos controladores P, PI, PD e PID, e além desses, também implementamos os filtros, para entender o impacto que eles causam nos sinais dos controladores. Nos experimentos 2A e 2B o objetivo era controlar os níveis dos tanques 1 e 2, respectivamente, e no experimento 2C era pedido que apenas o segundo tanque fosse controlado. Para analisar os gráficos observamos o tempo de estabilidade, o tempo de subida, o sobressinal e a oscilação.

Foi observado que no sistema de primeira ordem, de acordo com nossas simulações, o tempo de estabilidade é menor em relação ao de segunda ordem, e também oscila menos. Além disso, esperávamos encontrar resultados melhores com o controle em cascata, porém o tempo de estabilidade aumentou e também ficou mais instável, isso pode ser um problema de *tunning*, mas mesmo com pesquisa e vários testes empíricos não chegamos a um resultado realmente bom, que justifique o aumento de complexidade na implementação do controle em cascata numa planta industrial real, por exemplo.

Em todos os experimentos foi possível ver a ação proporcional, integral e derivativa, e as consequências de cada uma delas nos gráficos. Portanto, para diminuição de erro de regime, regula-se o K_i , para atuar na oscilação usamos o K_p e para tornar o tempo de acomodação menor é importante usar o K_d .

Referências

ARAÚJO, F. M. U. de. *Roteiro de laboratório 2A*. 2020. Disponível em: <<http://www.dca.ufrn.br/~meneghet>>. Acesso em: 20 set. 2020. Citado 3 vezes nas páginas 7, 8 e 10.

MADEIRA, D. *Controlador proporcional em sistemas de segunda ordem*. 2016. Disponível em: <<https://www.embarcados.com.br/controlador-proporcional-em-sistemas-de-segunda-ordem/>>. Acesso em: 01 nov. 2020. Citado na página 10.

OGATA, K. *Modern Control Engineering*. [S.l.]: Prentice Hall, 2001. Citado na página 6.

WESCOTT, T. *PID without a PhD*. 2020. Disponível em: <<http://manuals.chudov.com/Servo-Tuning/PID-without-a-PhD.pdf>>. Acesso em: 12 out. 2020. Citado na página 7.