

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS CORNÉLIO PROCÓPIO
DIRETORIA DE GRADUAÇÃO
CURSO SUPERIOR DE ENGENHARIA ELÉTRICA
GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

NATANEA APARECIDA ALVES

**ESTUDO E ANÁLISE DE TÉCNICAS DE CONTROLE CLÁSSICO E
MODERNO APLICADAS A SISTEMAS DE NÍVEL E TEMPERATURA
DE LÍQUIDOS**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2014

NATANEA APARECIDA ALVES

ESTUDO E ANÁLISE DE TÉCNICAS DE CONTROLE CLÁSSICO E MODERNO APLICADAS A SISTEMAS DE NÍVEL E TEMPERATURA DE LÍQUIDOS

Trabalho de Conclusão de Curso apresentada ao Curso Superior de Engenharia Elétrica da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de “Engenheiro Eletricista em Engenharia Industrial Elétrica”.

Orientador: Prof. Dr. Paulo Rogério Scalassara

Co-orientador: Prof. Dr. Cristiano Marcos Agulhari

CORNÉLIO PROCÓPIO

2014



Universidade Tecnológica Federal do Paraná
Campus Cornélio Procópio
Departamento de Engenharia Elétrica
Curso de Engenharia Industrial Elétrica



FOLHA DE APROVAÇÃO

Natanea Aparecida Alves

Estudo e análise de técnicas de controle clássico e moderno aplicadas a sistemas de nível e temperatura de líquidos

Trabalho de conclusão de curso apresentado às 16:00hs do dia 26/11/2014 como requisito parcial para a obtenção do título de Engenheiro Eletricista no programa de Graduação em Engenharia Industrial Elétrica da Universidade Tecnológica Federal do Paraná. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof(a). Dr(a). Paulo Rogério Scalassara - Presidente (Orientador)

Prof(a). Dr(a). Cristiano Marcos Agulhari - (Coorientador)

Prof(a). Dr(a). Emerson Ravazzi Pires da Silva - (Membro)

Prof(a). Dr(a). Wagner Endo - (Membro)

Pai e Mãe,
todos os agradecimentos que eu fizer aqui serão ainda insuficientes para quantificar o quanto sou grata, o quanto vocês foram importantes em todos esses anos de luta e sacrifício. Obrigada pelo amor, carinho, obrigada por serem meus pais. Amo vocês!!!

AGRADECIMENTOS

Quero agradecer acima de tudo à Deus pela força, saúde, fé e coragem que me destes durante toda esta longa caminhada e por proporcionar-me à graça de alcançar este grande sonho que se concretiza. Agradeço ainda por dar-me a melhor família do mundo, pois, sem o apoio, incentivo e dedicação de todos, eu não conseguiria chegar a esse dia tão especial. Dessa forma, agradeço imensamente à minha base e referência de tudo que tenho na vida, minha Família.

Mãe, obrigada pelo encorajamento durante todo esse processo e por ser a melhor mãe que alguém poderia desejar.

Pai, o senhor é a grande razão disso tudo, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir. Obrigada por ser minha fonte de orgulho e admiração.

Aos meus irmãos, Michele, Pedro Henrique e Naeane, que são meu ponto de apoio. A união e amor de todos vocês fizeram com que eu tivesse força para continuar lutando, e me tornar capaz de enfrentar novos desafios sabendo que vocês estarão sempre ao meu lado. Amo vocês mais que tudo!

Tão grande é meu agradecimento ao meus melhores e maiores presentes, Pedro Lucca e Pietra Luisa, que em muitos finais de semana me proporcionaram carinho e sorrisos tão lindos. Vocês são as razões da minha felicidade.

Agradeço em especial ao meu incrível sobrinho Pedro Luis, meu melhor amigo e companheiro de colchão na sala. "Você é a cobertura do meu sorvete, e sem cobertura, sorvete nenhum tem graça". Não existem palavras para descrever como é grande o meu amor por você!

Ao meu namorado Eric, obrigada pela compreensão do tempo de ausência, pelo estímulo nas horas de desânimo, pela atenção e carinho. Você tornou meus dias mais alegres e produtivos desde o dia em que o conheci. Obrigada por tudo amor!

Dedico essa conquista à vocês, os quais foram os grandes responsáveis pela conquista desse sonho, além do apoio inestimável em todos os momentos de dificuldades, acreditando que a realização desse sonho seria possível. Obrigada por tudo que fizeram por mim, pela oportunidade de realiza-lo e sobretudo, pela lição de amor que me ensinaram durante toda a vida.

Ao meu orientador Prof. Dr. Paulo Rogério Scalassara, agradeço-o de uma forma única e imensa. Mesmo com todas as minhas limitações e dificuldades, me recebeu e acreditou nesse trabalho. Obrigada pela paciência e excelente trabalho de orientação.

Ao Prof. Dr. Cristiano Marcos Agulhari, obrigada pelas imensas contribuições dadas ao longo deste trabalho.

Quero agradecer também a todos os colegas do laboratório do CIPECA, especialmente ao Murilo, Pedro, Thiago e Guilherme, vocês fizeram parte dessa conquista também.

E por fim, agradeço imensamente aos meus amigos, companheiros e integrantes do Bonde dos Caiporas, Gabriela, Carlos, Pereira, Igor e Marcelo. Muito obrigada pela amizade, paciência, momentos de risada, diversão e estudos compartilhados.

Entrego-me em vossas mãos poderosas, na certeza de que tudo vou alcançar.
(Oração da Fé)

RESUMO

ALVES, Natanea Aparecida. **Estudo e Análise de Técnicas de Controle Clássico e Moderno Aplicadas a Sistemas de Nível e Temperatura de Líquidos**. 2014. 73 f. Trabalho de Conclusão de Curso – Graduação em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2014.

Neste trabalho, propõe-se o estudo de uma planta composta por dois processos industriais: sistemas de controle de nível e temperatura de líquidos. Foram aplicadas duas técnicas distintas de controle para o sistema de nível com o intuito de comparação: uma baseada em controle clássico e a outra em controle moderno. Para o sistema de temperatura, foi implementado somente uma técnica de controle clássico a partir do seu sistema em malha aberta. E além disso, realizou-se um ensaio em malha fechada do controle da temperatura para o sistema completo, aonde o nível do reservatório de líquidos foi mantido constante e o suficiente para não danificar o aquecedor. A planta é composta por um reservatório em vidro com dois tanques e uma bomba CC. Estes processos são monitorados e controlados automaticamente através de uma interface Planta/Microcontrolador Arduino/Matlab. O objetivo deste trabalho é aplicar procedimentos de identificação, modelagem e controle para ambos os sistemas. Assim a partir da aquisição dos dados por esta interface, são implementados os modelos de identificação e, em seguida, as estratégias de controle para cada tipo de sistema. Apresenta-se com uma discussão do desempenho do controle clássico e do controle moderno baseada nos resultados obtidos. Embora foi possível realizar o controle dos sistemas com a metodologia proposta, mais testes ainda precisam ser realizados para melhorar o desempenho da solução.

Palavras-chave: Controle Clássico, Controle Moderno, Sistema de nível, Sistema de Temperatura, Microcontrolador Arduino

ABSTRACT

ALVES, Natanea Aparecida. **Study and Analysis of Classic and Modern Control Techniques Applied to Systems Level and Temperature of Liquids**. 2014. 73 f. Trabalho de Conclusão de Curso – Graduação em Engenharia Elétrica, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2014.

In this work, we propose the study of a plant which consists of two industrial process : control of liquid level and temperature systems. One based on classic control and the other on modern control: two different techniques for level system control with intention of comparison were applied. For the system temperature, only one technique of classical control was implemented from its open-loop system. And besides, there was a test in closed-loop temperature control for the entire system, where the liquid reservoir level was kept constant and enough to not damage the heater. The plant comprises an acrylic reservoir with two container and a pump DC. These processes are monitored and controlled automatically using a interface Plant / Arduino Microcontroller / Matlab. The objective of this work is to apply identifying procedures , modeling and controlling both systems. Thus from the data acquisition using the interface, are implemented the identified models and then strategies control for each type of system. Is presented a discussion of the performance of classical control and modern control based on the results obtained. Although it was possible to make the control of systems with the proposed methodology, more tests need to be conducted to further improve the performance of the solution.

Keywords: Classic Control, Modern Control, System Level, Temperature System, Arduino Microcontroller

LISTA DE FIGURAS

FIGURA 1	– Diagrama de blocos de um sistema de controle.	14
FIGURA 2	– Diagrama de blocos de sistema de controle em malha fechada.	15
FIGURA 3	– Diagrama de blocos de sistema de controle em malha aberta.	16
FIGURA 4	– Procedimento para identificação de processos.	16
FIGURA 5	– Curva com atraso de transporte.	18
FIGURA 6	– Curva sem atraso de transporte.	19
FIGURA 7	– Diagrama de blocos da função de transferência em malha fechada.	20
FIGURA 8	– Sistema de controle de malha fechada.	23
FIGURA 9	– Sistema de controle de malha fechada com bloco integrador.	24
FIGURA 10	– Diagrama de blocos de um controlador PID.	25
FIGURA 11	– Reservatório de líquido do Módulo de Nível e Temperatura 2325 Datapool.	27
FIGURA 12	– Distância que o sinal percorrerá para sair e retornar ao sensor.	28
FIGURA 13	– Sensor Ultrassônico HC-SR04.	28
FIGURA 14	– Diagrama de temporização do pino TRIG e ECHO.	29
FIGURA 15	– Pinagem do L298N.	30
FIGURA 16	– Ponte H.	30
FIGURA 17	– Arduino Uno.	31
FIGURA 18	– PT-100.	32
FIGURA 19	– Ponte de Wheatstone.	32
FIGURA 20	– Método de ligação a 2 fios.	33
FIGURA 21	– Passagem por zero rede elétrica.	34
FIGURA 22	– Tensão e Ângulo de disparo.	34
FIGURA 23	– Referência para a passagem por zero.	35
FIGURA 24	– Formas de onda de tensão na carga.	35
FIGURA 25	– Circuito detector de passagem por zero.	36
FIGURA 26	– Circuito de acionamento da resistência de aquecimento.	36
FIGURA 27	– Diagrama do controle do Sistema de Nível.	37
FIGURA 28	– Diagrama do controle do sistema de temperatura.	38
FIGURA 29	– Ensaio em malha aberta para o sistema de nível.	38
FIGURA 30	– Ensaio em malha aberta e função de transferência identificada.	39
FIGURA 31	– Velocidade do sistema de nível obtida pela derivada.	40
FIGURA 32	– Resposta em malha aberta para o sistema de nível.	41
FIGURA 33	– Resposta em malha aberta para variável velocidade.	41
FIGURA 34	– Ensaio em malha aberta para o sistema de temperatura.	43
FIGURA 35	– Sistema de Temperatura em malha aberta e identificada.	43
FIGURA 36	– Ensaio 1 - Controle PI para sistema de nível.	44
FIGURA 37	– Ensaio 2 - Controle PI para sistema de nível.	45
FIGURA 38	– Ensaio 3 - Controle PI para sistema de nível.	45
FIGURA 39	– Ensaio 1 - Alocação de polos.	46
FIGURA 40	– Ensaio 2 - Alocação de polos.	47
FIGURA 41	– Ensaio 3 - Alocação de polos.	47
FIGURA 42	– Comparação: a) Ensaio 1 - Controle PI. b) Ensaio 1 - Alocação de polos.	48
FIGURA 43	– Ensaio 1 - Controle por realimentação de estados.	48

FIGURA 44 – Comparação: a) Ensaio 2 - Controle PI. b) Ensaio 1 - Controle por realimentação de estados.	49
FIGURA 45 – Ensaio 2 - Controle por realimentação de estados.	49
FIGURA 46 – Ensaio 1 - Temperatura.	50
FIGURA 47 – Ensaio 2 - Temperatura.	51
FIGURA 48 – Ensaio 3 - Temperatura.	51

LISTA DE TABELAS

TABELA 1 – Método de Ziegler e Nichols	19
--	----

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVOS	12
1.2	ESTRUTURA	12
2	TEORIA DE CONTROLE	14
2.1	FUNDAMENTOS DE SISTEMAS DE CONTROLE	14
2.2	IDENTIFICAÇÃO DE SISTEMAS	16
2.2.1	Sistemas de Primeira Ordem	17
2.3	MODELAGEM NO ESPAÇO DE ESTADOS	21
2.3.1	Projeto do Controlador	23
2.4	SISTEMAS DE CONTROLE PID	24
2.4.1	Controlador Proporcional	25
2.4.2	Controlador Integral	26
2.4.3	Controlador Derivativo	26
3	MATERIAIS E MÉTODOS	27
3.1	MATERIAIS	27
3.1.1	Planta Didática	27
3.1.2	Sensor Ultrassônico HC-SR04	28
3.1.3	Módulo de acionamento L298N	29
3.1.4	Microcontrolador Arduino	30
3.1.5	Sensor Pt-100	31
3.1.6	Acionamento da resistência de aquecimento	33
3.2	METODOLOGIA	36
3.2.1	Controle Clássico para o Sistema de Nível	38
3.2.2	Controle Moderno para o Sistema de Nível	39
3.2.3	Controle Clássico para o Sistema de Temperatura	42
4	RESULTADOS E DISCUSSÕES	44
4.1	RESULTADOS DO SISTEMA DE NÍVEL	44
4.1.1	Controle Clássico	44
4.1.2	Controle Moderno	46
4.2	RESULTADOS DO SISTEMA DE TEMPERATURA	50
5	CONCLUSÕES	52
5.1	CONSIDERAÇÕES FINAIS	52
5.2	SUGESTÕES PARA TRABALHOS FUTUROS	53
	REFERÊNCIAS	54
	APÊNDICE A – CÓDIGO ARDUINO 1	56
	APÊNDICE B – CÓDIGO ARDUINO 2 - MATLAB	58
B.1	CÓDIGO DO CONTROLE CLÁSSICO PARA O SISTEMA DE NÍVEL	58
B.2	CÓDIGO DO CONTROLE CLÁSSICO PARA O SISTEMA DE TEMPERATURA	60
B.3	CÓDIGO DO CONTROLE MODERNO PARA O SISTEMA DE NÍVEL	62
	APÊNDICE C – CÓDIGO ARDUINO 3	66
	ANEXO A – CÓDIGO ARDUINO 2	68

1 INTRODUÇÃO

No setor industrial, a automação está ligada à mecanização e tem como objetivo principal a substituição do trabalho humano por máquinas ou processos controlados automaticamente. A finalidade é reduzir o esforço ou até mesmo a interferência humana no processo. Devido à constante evolução e aprimoramento das novas tecnologias, a automação movimenta e impulsiona o aumento da eficiência dos processos, qualidade, produtividade, flexibilidade, redução de custos e precisão para atender aos interesses dos clientes, cada vez mais exigentes (SCHMIDT, 2008).

Segundo Moraes e Castrucci (2001), automação se define quando qualquer sistema apoiado em computadores substitui o trabalho humano, visando a segurança das pessoas, qualidade dos produtos, rapidez ou redução de custos da produção, aperfeiçoando os complexos objetivos das indústrias. Além disso, as operações feitas automaticamente são mais eficientes e de maior qualidade do que as manuais, por isso, levando em consideração fatores econômicos e sociais, encontra-se a grande motivação para o desenvolvimento de sistemas automáticos.

As indústrias, que tem a necessidade de automatizar processos ou máquinas, buscam cada vez mais desenvolver projetos utilizando microcontroladores, pois em geral, é mais rápido e fácil. Sua programação permite aplicações em diversos tipos de controle industriais, pois são componentes extremamente flexíveis devido a sua programabilidade (SCHMIDT, 2008).

De modo geral, os microcontroladores aumentam a eficiência do processo, permitem redução de custo facilitando e melhorando a vida dos usuários e valorizando os produtos onde são aplicados e estão presentes nas mais diversas áreas, dentre as quais estão a automação industrial, comercial e predial, área automobilística, agrícola, produtos manufaturados, eletrodomésticos e telecomunicações (QUEIROGA; MONTEIRO; NETO, 2012).

Microcontroladores com seu tamanho reduzido e necessidade mínima de componentes externos, tornam-se ideais para a realização de tarefas de monitoramento remoto, tais como medição de temperatura e nível de líquidos de reservatórios (MENEZES et al., 2010).

Dessa forma, controlar o nível de um reservatório de líquidos é fundamental em diversos processos, sejam em industriais, residências, laboratórios, entre outros. Em ambientes industriais, muitos processos necessitam que os níveis de líquidos sejam mantidos nos patamares desejados para o sucesso da produção. Essas e outras aplicações justificam um estudo sobre processos automatizados para controle de níveis de líquidos. Se esse controle for eficiente, simples e de fácil manutenção, os benefícios são ainda maiores (SCHMIDT, 2008).

Da mesma maneira, a temperatura é sem dúvida uma variável importante nos processos industriais, e sua medição e controle são essenciais para a qualidade do produto e a segurança não só das máquinas como também do homem. O controle de temperatura é necessário em processos industriais, tais como fornos industriais, centrais de ar-condicionado, caldeiras, chocadeiras, sistemas químicos e uma infinidade de aplicações (FAATESP, 2014).

Um tipo de controlador bastante utilizado na indústria, segundo Carmo (2006), é o Controlador Proporcional, Integral e Derivativo (PID), dado sua robustez e facilidade de implementação. Esse controlador é projetado, normalmente, com a resposta de um modelo de referência e, a partir do conhecimento da planta controlada, encontra-se os parâmetros de ganho proporcional, integral e derivativo desse controlador (ASTROM; HAGGLUND, 1995).

Uma alternativa à tradicional técnica de Controle Clássico implementada neste tra-

balho, é a técnica de Controle Moderno por Realimentação de Estados, baseado no método da alocação de polos, que segundo Ogata (2010), este método é, de certa maneira, similar ao método do lugar das raízes, no qual alocamos os polos de malha fechada na posição desejada. A diferença básica é que, no projeto pelo lugar das raízes, alocamos somente os polos dominantes de malha fechada nas posições desejadas, enquanto no projeto por alocação de polos alocamos todos os polos de malha fechada nas posições desejadas. Outra diferença que deve-se levar em consideração é que no método de alocação de polos utiliza-se um ganho de realimentação de estados, enquanto o método do lugar das raízes baseia-se na realimentação da saída utilizando compensadores. A abordagem do *espaço de estados* (também conhecida como abordagem moderna ou no domínio do tempo) é um método unificado para modelar, analisar e projetar uma variedade de sistemas (NISE, 2012).

Para o sistema de nível, aplica-se a técnica do controle clássico e controle moderno, sendo desenvolvido um estudo comparativo dos desempenhos obtidos para cada tipo de controle. Para o sistema de temperatura, realiza-se o controle clássico, com a aplicação do Controlador Proporcional-Integral (PI). A aplicação da técnica de controle moderno para o sistema de temperatura não foi realizada devido ao sistema possuir apenas uma variável de estado, a temperatura, a qual apresenta muito ruído.

Dessa forma, realiza-se o controle de temperatura para o sistema completo, onde o nível do reservatório de líquidos será mantido constante e o suficiente para não danificar o aquecedor. Utiliza-se essa estratégia de modo a poder desconsiderar efeitos multivariáveis. Para ambos os sistemas, desenvolvem-se simulações, testes e implementações das técnicas. Por fim, será realizado o controle de nível e temperatura no reservatório de líquido simultaneamente para a comprovação do funcionamento dos sistemas de controle.

1.1 OBJETIVOS

Este trabalho tem como objetivo propor um procedimento de identificação e modelagem para um sistema de nível e de temperatura utilizando dois tanques conectados. Além disso, apresentar um estudo comparativo entre técnicas de controle clássico e controle moderno aplicados ao sistema de nível de líquidos, e somente controle clássico, aplicado ao sistema de temperatura utilizando um Controlador Proporcional-Integral (PI).

Para atingir o objetivo proposto neste trabalho, foram realizadas as seguintes etapas:

- Estudar e analisar as técnicas de controle para ambos os sistemas;
- Coletar e analisar os dados da função de transferência de cada sistema em malha aberta;
- Aplicar a técnica de controle clássico utilizando o controlador PI para o sistema de temperatura;
- Aplicar as técnicas de controle clássico e moderno para o sistema de nível de líquidos.

1.2 ESTRUTURA

Este trabalho está estruturado da seguinte maneira: no Capítulo 2, apresenta-se uma revisão bibliográfica sobre técnicas de controle utilizadas, como também as teorias relacionadas

ao desenvolvimento deste trabalho. Em seguida, no Capítulo 3, apresentam-se os materiais utilizados, além dos métodos e dos ensaios propostos. Na sequência, no Capítulo 4, os principais resultados dos ensaios realizados, bem como as discussões. Por fim, no Capítulo 5, as considerações finais deste trabalho.

2 TEORIA DE CONTROLE

Neste capítulo, serão apresentados alguns conceitos abordados no trabalho, com uma revisão bibliográfica sobre Sistemas de Controle, identificação de sistemas, método de Controle Clássico com a utilização do controlador PI e método de Controle Moderno com realimentação de variáveis de estado usando o método de alocação de polos.

2.1 FUNDAMENTOS DE SISTEMAS DE CONTROLE

Segundo Nise (2012), um sistema de controle consiste em subsistemas e processos (ou plantas) construídos com o objetivo de obter uma saída desejada com um desempenho desejado, dada uma entrada específica. Um sistema pode ser representado pelo diagrama de blocos da Figura 1.

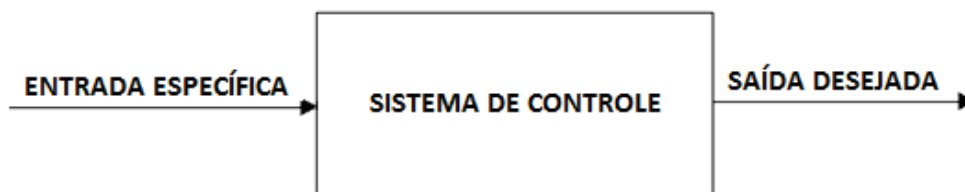


Figura 1 – Diagrama de blocos de um sistema de controle.

Fonte: Adaptado de Nise (2012)

Dessa forma, Ogata (2010) orienta que é necessário possuir o conhecimento de alguns conceitos básicos bem definidos sobre sistemas de controle para realizar uma análise ou mesmo o controle de um sistema.

- **Processo:** desenvolvimento caracterizado por uma série de modificações graduais que se sucedem umas às outras de modo relativamente estável, avançando em relação a dado resultado ou objetivo.
- **Sistemas:** é a combinação de componentes que agem em conjunto para atingir determinado objetivo.
- **Plantas:** pode ser uma parte de equipamento ou apenas um conjunto de componentes de um equipamento que funcione de maneira integrada, com o objetivo de realizar determinada operação.
- **Variável controlada:** é a grandeza ou a condição que é medida e controlada, normalmente sendo a saída do sistema.
- **Variável manipulada:** é a grandeza ou a condição modificada pelo controlador, de modo que afete o valor da variável controlada.
- **Distúrbios:** é um sinal que tende a afetar de maneira adversa o valor da variável de saída de um sistema. Se um distúrbio for gerado dentro de um sistema, ele será chamado

distúrbio interno, enquanto um distúrbio externo é aquele gerado fora do sistema e que se comporta como um sinal de entrada do sistema.

- Controle: significa medir o valor da variável controlada do sistema e aplicar um sinal de controle ao sistema para corrigir ou limitar os desvios do valor medido a partir de um valor desejado.
- Controle com realimentação: refere-se a uma operação que, na presença de distúrbios, tende a diminuir a diferença entre a saída do sistema e alguma entrada de referência e atua com base nessa diferença.

Após a definição desses conceitos definem-se as duas principais configurações dos sistemas de controle: malha fechada e malha aberta.

- Malha Fechada: Em um sistema de controle de malha fechada, o sinal de erro atuante é a diferença entre o sinal de entrada e o sinal de realimentação (podendo ser o próprio sinal de saída). Portanto, a realimentação do controlador é realizada de modo a minimizar o erro e acertar a saída do sistema ao valor desejado como mostra a Figura 2 (OGATA, 2010).

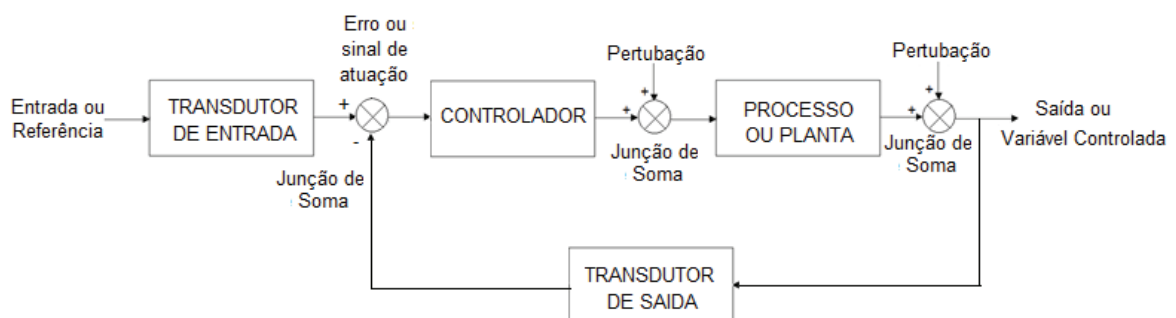


Figura 2 – Diagrama de blocos de sistema de controle em malha fechada.

Fonte: Adaptado de Nise (2012)

- Malha Aberta: Em um sistema de controle de malha aberta, o sinal de saída não é medido nem realimentado para comparação com a entrada (OGATA, 2010), como mostra a Figura 3. Ele começa com um subsistema chamado de transdutor de entrada, a qual converte a forma de entrada para aquela utilizada pelo controlador. O controlador aciona um processo ou uma planta. A entrada algumas vezes é chamada de referência, enquanto a saída pode ser chamada de variável controlada. Outros sinais, como as perturbações, são mostrados adicionados às saídas do controlador e do processo através de junções soma, as quais fornecem a soma algébrica dos seus sinais de entrada utilizando os sinais associados (NISE, 2012).

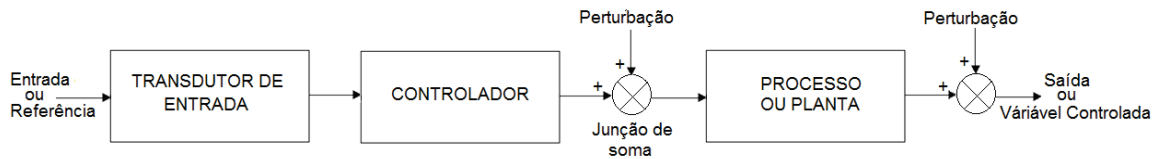


Figura 3 – Diagrama de blocos de sistema de controle em malha aberta.

Fonte: Adaptado de Nise (2012)

2.2 IDENTIFICAÇÃO DE SISTEMAS

A identificação de sistemas consiste na determinação de um modelo matemático que represente os aspectos essenciais do sistema, caracterizado pela manipulação dos sinais de entrada e saída e que estão relacionados através de uma função de transferência contínua ou discreta. Em Aguirre (2007), define-se modelagem matemática como a área do conhecimento que estuda maneiras de construir e implementar modelos (matemáticos) de sistemas reais, sendo fundamental que o modelador matemático saiba que o modelo desenvolvido para um determinado sistema é apenas uma representação aproximada de algumas características do sistema.

Uma vez obtido o modelo matemático de um sistema, podem ser utilizadas várias ferramentas analíticas e computacionais para efeito de análise e síntese. Para análise de resposta transitória ou da resposta em frequência de um sistema linear, invariante no tempo, de entrada e saída única, a representação pela função de transferência pode ser mais conveniente do que qualquer outra (OGATA, 2010).

Segundo Coelho e Coelho (2004), o modelo matemático final é uma forma de conhecimento da relação existente entre os sinais de entrada e saída, caracterizada no processo físico pela função de transferência, com isto, nos orienta que na tarefa de identificação, há diferentes procedimentos para geração do sinal de entrada, medição da saída e armazenamento dos dados. A Figura 4 ilustra a composição básica de um processo de identificação.

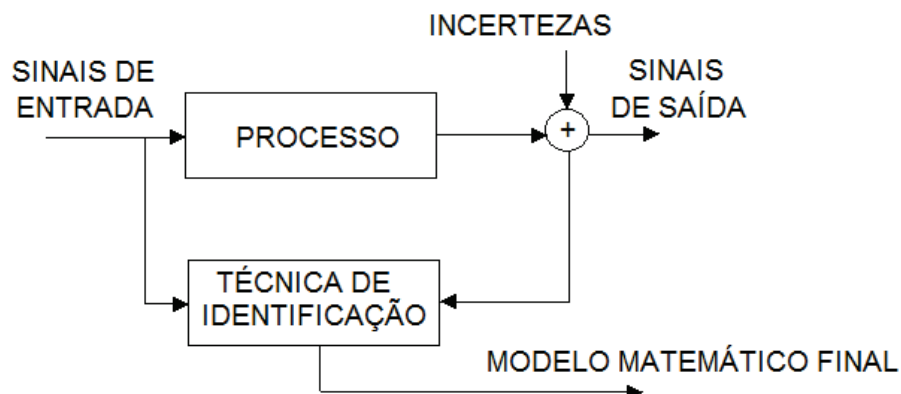


Figura 4 – Procedimento para identificação de processos.

Fonte: Coelho e Coelho (2004)

A identificação de sistemas se propõe a obter um modelo matemático que explique de forma aproximada, a relação de causa e efeito presente nos dados. Assim, as principais etapas

de um problema de identificação são (AGUIRRE, 2007):

- Identificação de um processo pelo teste de resposta ao degrau: o teste de resposta ao degrau, por natureza, não permite a estimação de modelos de ordem superior, já que o sinal degrau tem uma pobre composição em frequência. Com a curva de reação do processo, é possível aplicar diferentes técnicas gráficas, numéricas ou computacionais para modelar o sistema controlado por funções de transferência de primeira e segunda ordem.
- Identificação pelo teste da resposta em frequência: o processo é submetido a uma entrada harmônica (sinal senoidal). De acordo com as curvas de magnitude e fase, é possível identificar as frequências de corte (avaliando-se a influência dos zeros e polos) e, conseqüentemente a função de transferência.
- Testes dinâmicos e coleta de dados. Uma vez que a identificação se propõe a obter modelos a partir de dados, é necessário gerar tais dados.
- Escala da representação matemática a ser usada. Funções de transferência em tempo contínuo são usadas em problemas de identificação determinística.
- Determinação da estrutura do modelo. No caso de modelos lineares, a escolha da sua estrutura se restringe, basicamente, à escolha do número de polos e zeros, bem como a determinação do atraso puro de tempo.
- Estimação de parâmetros. Essa etapa começa com a escolha do algoritmo a ser utilizado.
- Validação do modelo. Tendo obtido uma família de modelos, é necessário verificar se eles incorporam ou não as características de interesse do sistema original.

Técnicas alternativas de modelagem matemática são estudadas através da identificação de sistemas. Nessas técnicas, os tipos de modelos e os requisitos são bastante distintos dos análogos na modelagem pela natureza do processo, além do fato de que pouco ou nenhum conhecimento prévio do sistema é necessário, o que torna a identificação de sistema, muitas vezes, preferível (AGUIRRE, 2007).

Para processos industriais, a partir de experimentos (procedimentos estatísticos, filtragem dados, etc.) é possível obter dados para o desenvolvimento de modelos matemáticos. O modelo final é representado pela função de transferência, uma forma do conhecimento da relação existente entre os sinais de entrada e saída (COELHO; COELHO, 2004).

2.2.1 Sistemas de Primeira Ordem

A partir dos modelos, temos as funções de transferência que revelam os termos que descrevem a resposta do processo, como ganho, constante de tempo e tempo de atraso (SMITH; CORRIPIO, 1997).

Segundo Coelho e Coelho (2004), para a identificação das características essenciais de um processo sob avaliação, é possível utilizar, por exemplo, uma entrada degrau. Assim, as características do sistema podem ser obtidas a partir do conhecimento da resposta ao degrau.

A representação usual de um modelo matemático para uma planta industrial é a função de transferência de primeira ordem com atraso de transporte, Equação (1), onde s é o operador Laplace, K é o ganho, θ é o atraso de transporte contínuo, τ é a constante de tempo, e $Y(s)$ e $U(s)$ são as transformadas de Laplace da saída e da entrada (COELHO; COELHO, 2004).

$$G_{p1}(s) = \frac{Y(s)}{U(s)} = \frac{Ke^{-\theta s}}{\tau s + 1} \quad (1)$$

Dessa forma, para a identificação do ganho em regime K é necessário utilizar a Equação (2), onde $\Delta y(t)$ é a variação da saída e $\Delta u(t)$ é a variação da entrada (COELHO; COELHO, 2004).

$$K = \frac{\Delta y(t)}{\Delta u(t)} \quad (2)$$

Os parâmetros (K, θ, τ) que caracterizam o comportamento dinâmico de um sistema de primeira ordem, podem ser identificados com base na resposta de reação do processo. A seguir apresenta-se o método de Ziegler/Nichols para se obter tais parâmetros, conforme Coelho e Coelho (2004).

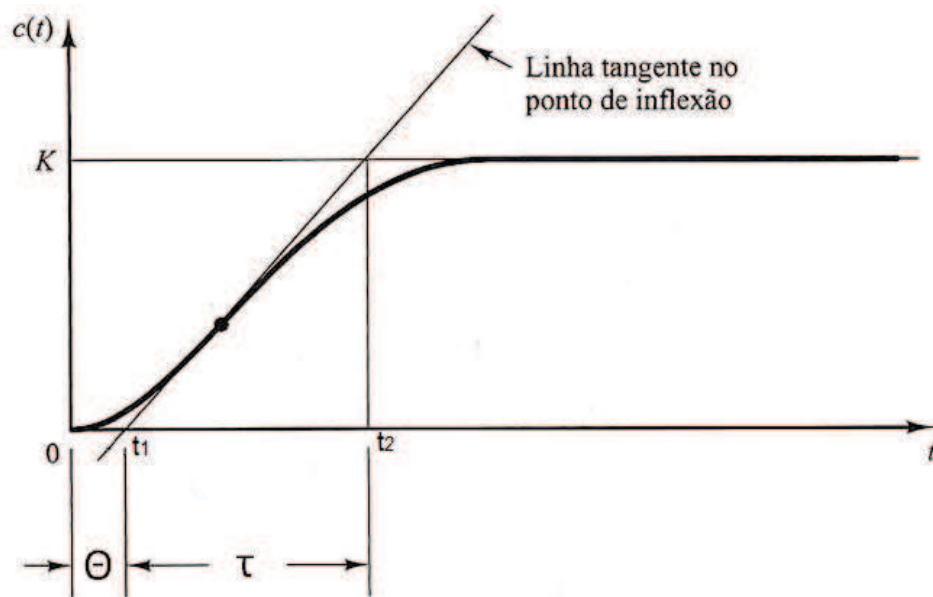


Figura 5 – Curva com atraso de transporte.

Fonte: Adaptado de Ogata (2010)

A reta traçada corresponde à tangente no ponto de máxima inclinação da curva de reação, Figura 5. O atraso θ é obtido através do cálculo do intervalo de tempo entre a aplicação do degrau na entrada e o instante t_1 em que a reta tangente toca a reta $c(t) = c(0)$, Equação (3).

$$\theta = t_1 \quad (3)$$

A constante de tempo τ é determinada pelo intervalo e tempo entre t_1 e o instante t_2

em que a reta tangente toca a reta $c(t) = K$, Equação (4).

$$\tau = t_2 - t_1 \quad (4)$$

Dessa forma, segundo Ziegler e Nichols, os valores de K_p , T_i e T_d possíveis para o controle utilizando o controlador PID podem ser sintonizados conforme a Tabela 1.

Tabela 1 – Método de Ziegler e Nichols

Tipo de Controlador	K_p	T_i	T_d
P	$\frac{\tau}{\theta}$	∞	0
PI	$0,9 \frac{\tau}{\theta}$	$\frac{\theta}{0,3}$	0
PID	$1,2 \frac{\tau}{\theta}$	2θ	$0,5\theta$

Fonte: Adaptado de Ogata (2010)

Caso o sistema em malha aberta apresente uma resposta ao degrau unitário com o aspecto da Figura 6, pode-se utilizar uma aproximação de primeira ordem sem atraso de transporte, como mostra a Equação (5).

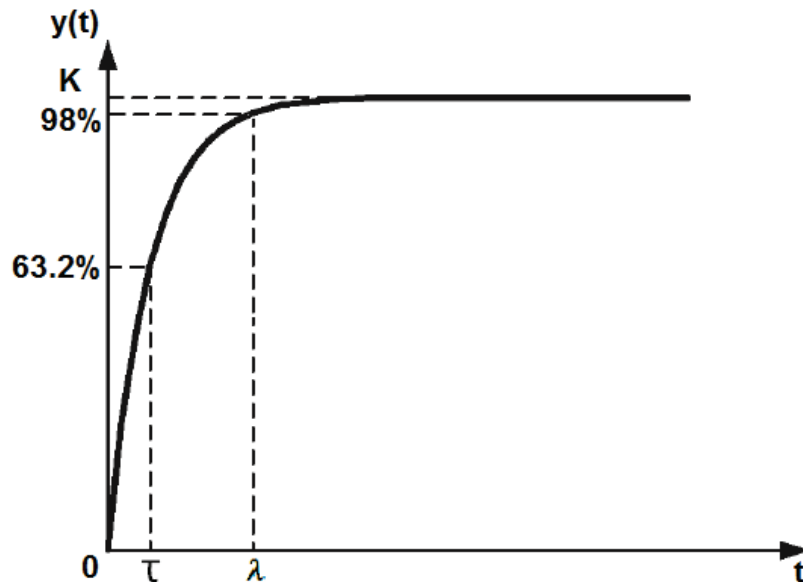


Figura 6 – Curva sem atraso de transporte.

Fonte: Autoria própria

$$G(s) = \frac{K}{\tau s + 1} \quad (5)$$

Nesse caso, um controlador PI é suficiente para fazer com que a resposta em malha fechada ao degrau unitário tenha erro estacionário nulo e constante de tempo λ .

Dessa maneira, os parâmetros da função de transferência podem ser identificados conforme descrito abaixo.

1. Ganho de regime: o ganho do sistema é obtido pela Equação (2).
2. Constante de tempo τ : obtida pelo tempo medido a partir do ponto onde o sinal de saída atinja 63,2% do seu valor (COELHO; COELHO, 2004).
3. Constante de tempo λ : obtida no momento em que a saída do processo atinja 98% do seu valor.

Nota-se, na Figura 7, que assumindo realimentação unitária com $G_c(s)$ sendo a função de transferência do controlador, $G(s)$ a função de transferência do sistema em malha aberta, obtemos a função de transferência de malha fechada do sistema $T(s) = Y(s)/X(s)$, conforme Equação (6).

$$T(s) = \frac{G(s)G_c(s)}{1 + G(s)G_c(s)} \quad (6)$$

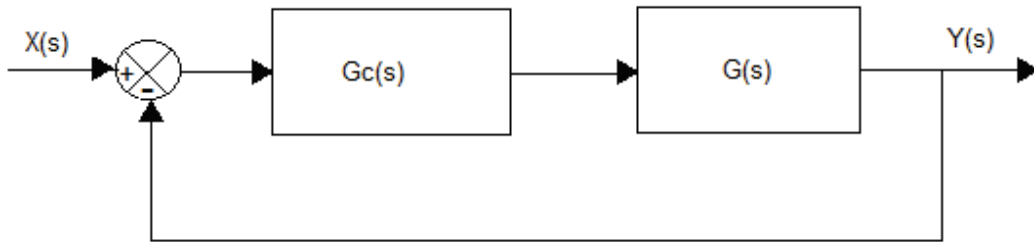


Figura 7 – Diagrama de blocos da função de transferência em malha fechada.

Fonte: Autoria própria

Isolando $G_c(s)$, tem-se:

$$G_c(s) = \frac{1}{G(s)} \frac{T(s)}{1 - T(s)}$$

Assim, para $T(s) = 1/(\lambda s + 1)$, verifica-se que

$$G_c(s) = \frac{\tau s + 1}{K} \frac{\frac{1}{\lambda s + 1}}{\left[1 - \frac{1}{\lambda s + 1}\right]}$$

Portanto, a função de transferência do controlador é obtida pela Equação (9).

$$G_c(s) = \frac{\tau s + 1}{K\lambda s} = \frac{\tau}{K\lambda} \left[1 + \frac{1}{\tau s}\right] \quad (9)$$

Dessa forma, verifica-se que a solução pode ser um Controlador PI conforme K_p e T_i , como mostra as Equações (10) e (11), respectivamente.

$$K_p = \frac{\tau}{K\lambda} \quad (10)$$

$$T_i = \tau \quad (11)$$

2.3 MODELAGEM NO ESPAÇO DE ESTADOS

Segundo Ogata (2010), a tendência atual dos sistemas de engenharia é no sentido de aumentar sua complexidade em função da necessidade de realizar tarefas complexas e com requisitos de boa precisão. Sistemas complexos podem ter muitas entradas, muitas saídas e podem ser variantes no tempo. A necessidade de satisfazer requisitos cada vez mais rigorosos quanto ao desempenho de sistemas de controle, facilitou o desenvolvimento da teoria de controle moderno, iniciada por volta de 1960. A teoria de controle moderno é uma abordagem centrada essencialmente no domínio do tempo, enquanto a teoria de controle convencional adota um enfoque no domínio de frequência.

Antes de prosseguir, conforme (OGATA, 2010), devem ser definidos os termos: estado, variáveis de estado, vetor de estado e espaço de estados.

- Estado: é o menor conjunto de valores de variáveis de estado, de modo que o conhecimento destes valores, junto com o conhecimento dos valores do sinal de entrada, determinam o comportamento do sistema.
- Variáveis de estado: constituem o menor conjunto de variáveis capaz de determinar o estado de um sistema dinâmico.
- Vetor de estado: vetor composto pelas n variáveis de estado necessárias para descrever completamente o comportamento de dado sistema, então essas n variáveis de estado poderão ser consideradas os n componentes de um vetor x , e este vetor é chamado vetor estado.
- Espaço de estados: o espaço n -dimensional, cujos eixos coordenados são formados pelos eixos de x_1, x_2, \dots, x_n , onde x_1, x_2, \dots, x_n são as variáveis de estado, é denominado espaço de estados. Qualquer estado pode ser representado por um ponto no espaço de estados.

A análise no espaço de estados envolve três variáveis na modelagem de sistemas dinâmicos: variáveis de entrada, variáveis de saída e variáveis de estado. A representação de um dado sistema no espaço de estados não é única, mas o número de variáveis de estado é o mesmo para qualquer uma das diferentes representações do mesmo sistema, no espaço de estados (OGATA, 2010).

Para se definir a equação de estado e de saída, segundo Dorf e Bishop (2001), deve ser definido inicialmente o vetor de estado, Equação (12), onde o negrito indica um vetor.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad (12)$$

O sinal de controle (escalar) é definido como u , dessa forma, o sistema pode ser representado através de notação compacta da Equação (13), chamada de equação de estado. Considera-se um sistema de uma entrada e uma saída.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (13)$$

A matriz \mathbf{A} $n \times n$ é uma matriz quadrada e \mathbf{B} é uma matriz $n \times 1$. A equação diferencial de estado relaciona a taxa de variação do estado do sistema ao próprio estado do sistema e aos sinais de entrada. Assim, as saídas de um sistema linear podem ser relacionadas com as variáveis de estado e com os sinais de entrada conforme Equação (14), chamada de equação de saída, aonde y é o sinal de saída expresso na forma escalar.

$$y = \mathbf{C}\mathbf{x} + Du \quad (14)$$

Segundo Nise (2012), para converter a representação no espaço de estados para uma função de transferência, aplica-se a transformada de Laplace nas equações de estado e de saída, admitindo condições iniciais nulas:

$$s\mathbf{X}(s) = \mathbf{A}\mathbf{X}(s) + \mathbf{B}U(s)$$

$$Y(s) = \mathbf{C}\mathbf{X}(s) + \mathbf{D}U(s)$$

Resolvendo para $\mathbf{X}(s)$, tem-se

$$(s\mathbf{I} - \mathbf{A})\mathbf{X}(s) = \mathbf{B}U(s)$$

ou

$$\mathbf{X}(s) = (s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}U(s)$$

em que \mathbf{I} é a matriz identidade.

Por substituição, tem-se a Equação (19).

$$Y(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}U(s) + \mathbf{D}U(s) = [\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}]U(s) \quad (19)$$

A matriz $[\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}]$ chama-se matriz de função de transferência, uma vez que ela relaciona a saída $Y(s)$ com a entrada $U(s)$. Dessa forma, pode-se obter a função de transferência, como mostra a Equação (20).

$$T(s) = \frac{Y(s)}{U(s)} = [\mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}] \quad (20)$$

A partir do momento que se obtém o modelo matemático dos componentes do sistema de controle, deve-se examinar as características desse sistema. Dessa forma, assegurar a estabilidade de um sistema de controle de malha fechada é uma questão central no projeto de sistemas de controle, e um sistema estável deve apresentar uma saída limitada se a entrada correspondente for limitada. Logo, a estabilidade de um sistema com retroação se relaciona diretamente com a localização das raízes da equação característica da função de transferência

do sistema (DORF; BISHOP, 2001).

2.3.1 Projeto do Controlador

Conforme Ogata (2010), se o sistema for considerado completamente controlável, então seus polos de malha fechada poderão ser alocados em qualquer posição desejada por meio de uma realimentação de estado, empregando uma matriz de ganho apropriada.

Contudo, existe um custo associado à alocação de todos os polos de malha fechada, porque essa alocação requer que todas as variáveis de estado possam ser medidas com sucesso, ou então, requer a inclusão de um observador de estado no sistema (OGATA, 2010).

Considerando as Equações (13) e (14), escolhe-se o sinal de controle como mostra a Equação (21), sendo r um sinal de referência na entrada e \mathbf{K} é o ganho de realimentação. O diagrama de blocos desse sistema é mostrado na Figura (8).

$$u = r - \mathbf{K}\mathbf{x} \quad (21)$$

Isso significa que o sinal de controle u é determinado por um estado instantâneo. Esse esquema é denominado realimentação de estados, onde a matriz \mathbf{K} $1 \times n$ é denominado matriz de ganho de realimentação de estado.

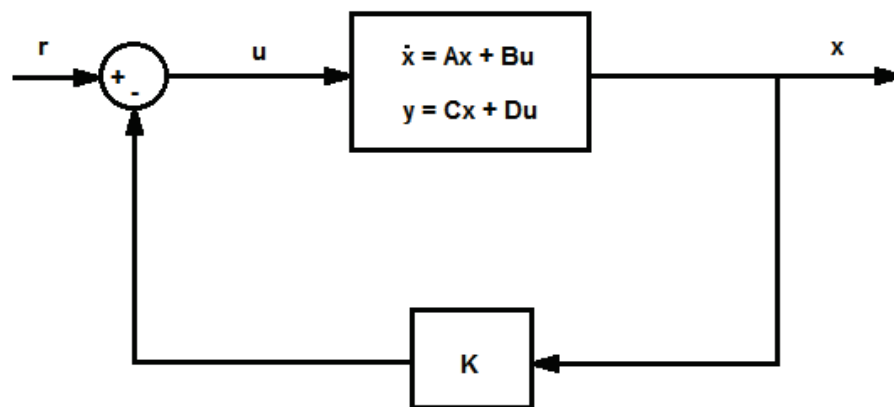


Figura 8 – Sistema de controle de malha fechada.

Fonte: Adaptado de Ogata (2010)

Na Equação (22), mostra-se como é realizada a realimentação dos estados para o sistema de controle de nível de líquidos.

$$u = r - \mathbf{K}\mathbf{x} = r - \begin{bmatrix} K_1 & K_2 \end{bmatrix} \begin{bmatrix} N_{ivel} \\ V_{elocidade} \end{bmatrix} = r - K_1 N_{ivel} - K_2 V_{elocidade} \quad (22)$$

Sendo as variáveis de estado para este sistema o N_{ivel} e a $V_{elocidade}$.

Contudo, para este trabalho, será adicionado ao sistema de controle um bloco integrador, de modo a minimizar o erro estacionário, como pode ser visto na Figura 9.

Para incluir o integrador ao sistema, considera-se que a integral do erro seja um dos estados do sistema. Desse modo, o sinal de controle aplicado é dado pela Equação (23). Assim,

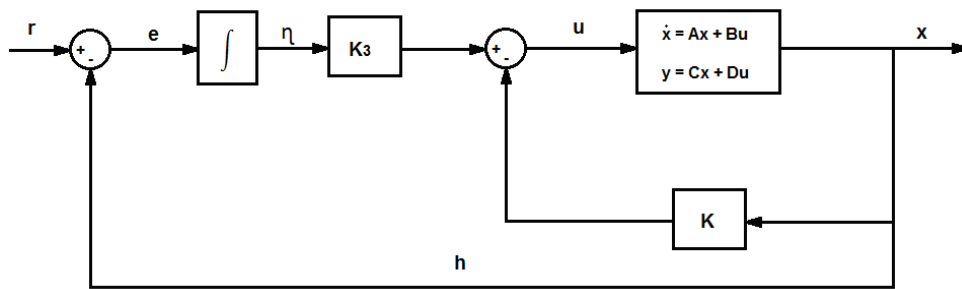


Figura 9 – Sistema de controle de malha fechada com bloco integrador.

Fonte: Adaptado de Ogata (2010)

faz-se necessário a obtenção de três polos em malha fechada, para obter o ganho de realimentação \mathbf{K} aplicado ao sistema.

$$u = -\mathbf{K}\mathbf{x} + K_3\eta = - \begin{bmatrix} K_1 & K_2 & -K_3 \end{bmatrix} \begin{bmatrix} N_{ivel} \\ V_{elocidade} \\ \eta \end{bmatrix} \quad (23)$$

onde $e = r - h$ e h a leitura do nível do reservatório, obtem-se η a partir da Equação (24).

$$\eta = \int_0^t e(\beta) d\beta \quad (24)$$

Tem-se na Equação (25) a representação no espaço de estados do sistema aumentado em malha fechada.

$$\frac{d}{dt} \begin{bmatrix} N_{ivel}(t) \\ V_{elocidade}(t) \\ \eta(t) \end{bmatrix} = \begin{bmatrix} A + B \begin{bmatrix} K_1 & K_2 \end{bmatrix} & -B[K_3] \\ \begin{bmatrix} -1 & 0 \end{bmatrix} & 0 \end{bmatrix} \begin{bmatrix} N_{ivel}(t) \\ V_{elocidade}(t) \\ \eta(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} r(t) \quad (25)$$

A definição da escolha dos valores do ganho de realimentação \mathbf{K} será explicada com mais detalhes no Capítulo Materiais e Métodos.

2.4 SISTEMAS DE CONTROLE PID

A utilidade dos controladores PID está na sua aplicabilidade geral à maioria dos sistemas de controle. Em particular, quando o modelo matemático da planta não é conhecido e, portanto, métodos de projeto analítico não podem ser utilizados, controles PID se tornam mais úteis (OGATA, 2010). Este controlador tem uma função de transferência dada pela Equação (26), sendo K_p , K_I e K_D os ganhos proporcional, integral e derivativo respectivamente (DORF; BISHOP, 2001).

$$G_c(s) = K_p + \frac{K_I}{s} + K_D s \quad (26)$$

Na equação (27), observa-se a saída do controlador no domínio do tempo $u(t)$.

$$u(t) = K_p e(t) + K_I \int_0^t e(t) dt + K_D \frac{de(t)}{dt} \quad (27)$$

O controlador oferece um termo proporcional que reduz o tempo de acomodação, o termo de integração diminui o erro de regime e o termo derivativo aumenta a estabilidade do sistema (DORF; BISHOP, 2001). Os ganhos Proporcional, Integral e Derivativo atuando em um sistema de controle são mostrados na Figura 10.

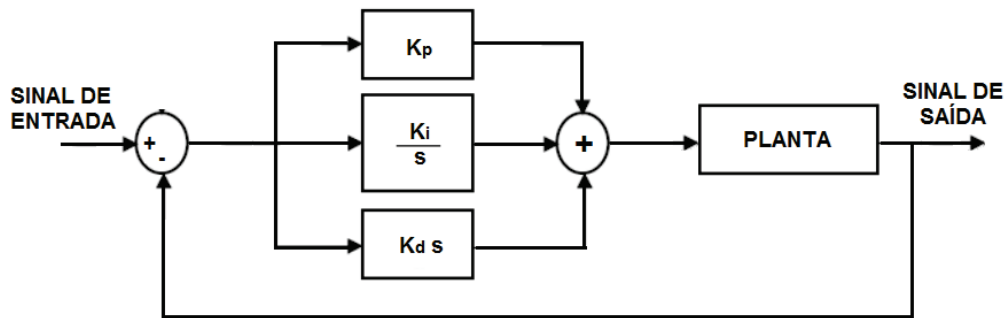


Figura 10 – Diagrama de blocos de um controlador PID.

Fonte: Autoria própria

O objetivo deste controlador é aproveitar as características particulares de cada uma destas ações a fim de obter uma melhora significativa do comportamento transitório e em regime permanente do sistema controlado (BEZERRA, 2010). Neste trabalho utiliza-se somente o controlador PI.

2.4.1 Controlador Proporcional

Nesse controlador, o sinal de controle é calculado com base no erro medido. Esse erro é a diferença entre o valor desejado e o valor medido da variável controlada, Equação (28).

$$e(t) = y_{ref} - y(t) \quad (28)$$

Existe uma relação linear entre o sinal de erro da entrada e saída da malha, assim a saída do processo terá uma resposta proporcional ao sinal de comando do atuador tentando estabilizar o sistema. Saída do controlador proporcional é dada pela Equação (29).

$$u(t) = K_p e(t) \quad (29)$$

Uma desvantagem do controle proporcional é que, para altos valores de K_p , o sistema pode se tornar instável, além de que, na presença de perturbações, não é possível obter erro de regime permanente nulo (DATAPOOL, 2003a).

2.4.2 Controlador Integral

O controlador integral puro tem ação de controle $u(t)$ proporcional a integral do erro. Conforme Equação (30), é possível determinar o controle integral.

$$u(t) = K_I \int_0^t e(t) dt \quad (30)$$

A desvantagem do uso do integrador no controlador é que o sistema fica mais lento, pois a resposta dependerá da acumulação do sinal de erro na entrada. Assim, como não é necessário um sinal de entrada para haver saída no controlador, pode-se reduzir ou até eliminar erros de estado estacionário (DATAPOOL, 2003a).

2.4.3 Controlador Derivativo

A ação derivativa nunca é utilizada de forma isolada, mas sempre em conjunto com as ações proporcional ou integral, pois, no caso de um erro nulo ou constante, a saída do controlador não irá apresentar nenhuma variação do sinal de entrada. A Equação (31) determina o controle derivativo.

$$u(t) = K_D \frac{de(t)}{dt} \quad (31)$$

A vantagem deste termo no controle é a velocidade de resposta, que se deve à imediata reação do diferenciador. E a desvantagem é que esse mesmo diferenciador é muito susceptível a ruídos de alta frequência, o que pode levar a distúrbios durante o processo de controle (DATAPOOL, 2003a).

3 MATERIAIS E MÉTODOS

Neste capítulo, apresentam-se os materiais utilizados neste trabalho. Na sequência, apresenta-se o esquema de controle utilizado tanto para o sistema de nível quanto para o sistema de temperatura, seguido pela apresentação dos ensaios que foram realizados.

3.1 MATERIAIS

3.1.1 Planta Didática

Sistemas de controle de nível e sistemas de controle de temperatura são frequentemente usados em processos e equipamentos industriais, e nesses processos, grandezas físicas necessitam ser controladas visando obter características operativas específicas. Neste projeto, o controle de nível é composto por dois tanques que se comunicam através de dutos que tem sua vazão controlada. Estes tanques fazem parte do Módulo de Nível e Temperatura 2325 da fabricante Datapool, Figura 11.



**Figura 11 – Reservatório de líquido do Módulo de Nível e Temperatura 2325 Datapool.
Fonte: Datapool (2003b)**

3.1.2 Sensor Ultrassônico HC-SR04

O sensor HC-SR04 permite detectar objetos com distâncias mínimas de 2 cm podendo chegar a distâncias máximas de até 5 m com uma precisão de 0,3 cm. Este sensor emite um sinal ultrassônico que reflete no objeto e retorna ao sensor. Este retorno é captado, permitindo-se calcular a distância do objeto ao sensor tomando o tempo de trânsito do sinal. A velocidade do sinal ultrassônico é de aproximadamente 340 m/s na água, assim, se o sensor estiver a uma distância d do objeto, o sinal percorrerá uma distância equivalente de $2d$ para sair e retornar ao sensor, conforme Figura 12.

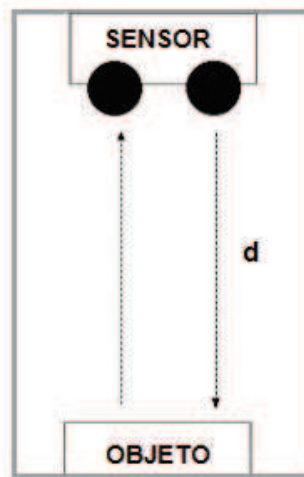


Figura 12 – Distância que o sinal percorrerá para sair e retornar ao sensor.

Fonte: Autoria própria

O sensor HC-SR04 possui 4 pinos de conexão, VCC, TRIG, ECHO e GND, sendo eles: alimentação de 5 V, pino de gatilho, pino de eco e terra respectivamente, Figura 13.

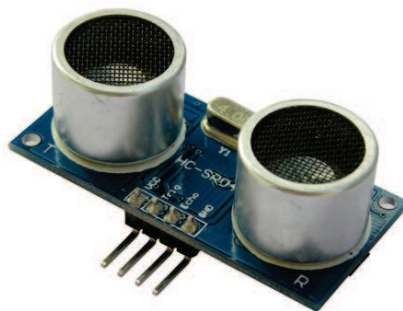


Figura 13 – Sensor Ultrassônico HC-SR04.

Fonte: Cytron Technologies (2013)

O pino TRIG normalmente deve estar em nível baixo, então, para o início da leitura da distância, este pino deve ser colocado em nível alto por pelo menos $10 \mu s$ e, em seguida, retornar para nível baixo. Neste momento, 8 pulsos de 40 kHz são emitidos e no pino ECHO será gerado um sinal em nível alto proporcional à distância do sensor ao objeto, como pode ser observado na Figura 14.

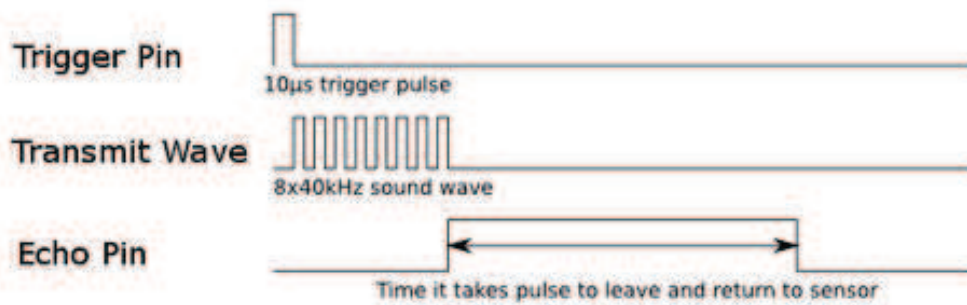


Figura 14 – Diagrama de temporização do pino TRIG e ECHO.

Fonte: Cytron Technologies (2013)

Assim, para obter a distância em centímetros, basta verificar o tempo em que o pino ECHO permaneceu em nível alto, e aplicar na Equação (32).

$$d_{centimetros} = \frac{ECHO_{tempo_microsegundos}^{alta}}{58} \quad (32)$$

Segundo Pelz, Junior e Vargas (2014), para converter a distância entre o sensor e a água para nível de água do reservatório, utiliza-se a Equação (33) onde, $N_{reservatorio}$ é o nível do reservatório em centímetros, d é a distância entre o sensor e a água, considerando 20 centímetros a distância entre o sensor e o fundo do reservatório.

$$N_{reservatorio} = 20 - d_{centimetros} \quad (33)$$

Ou seja, a equação do nível do reservatório que será utilizado no trabalho, decorre da relação inversamente proporcional da distância entre o sensor e o nível da água em relação ao fundo do reservatório.

3.1.3 Módulo de acionamento L298N

O CI L298N, Figura 15, é um circuito integrado no encapsulamento multiwatt 15 que possui internamente duas pontes H. Ele é um driver de acionamento de alta tensão e corrente, podendo operar com até 46 V de alimentação e conduzir até 2 A (por canal) para a carga. Projetado para níveis lógicos padrão TTL e para acionar cargas indutivas e motores CC, tendo duas entradas de habilitação (EnA e Enb) são fornecidas para ativar ou desativar o dispositivo, independente dos sinais de entrada (Input1, Input2, Input3, Input4) (ST MICROELECTRONICS, 2000).

Para alterar o sentido da rotação do motor, basta apenas ligar os terminais do motor de forma invertida. Para que não seja necessário fazer essa operação manualmente, pode-se utilizar uma ponte H. Como pode ser observada na Figura 16, uma ponte H básica é composta por 4 chaves mecânicas ou eletrônicas posicionadas formando a letra H, sendo que cada uma localiza-se num extremo e o motor é posicionado no meio. Pode-se criá-la facilmente com a finalidade de controlar o sentido da rotação de um motor utilizando chaves simples, relés ou transistores (ARDUINO, 2014).

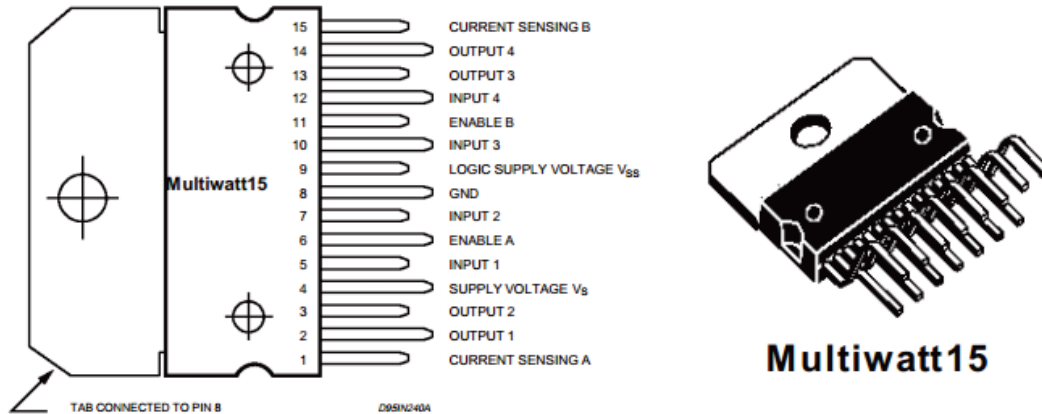


Figura 15 – Pinagem do L298N.

Fonte: Adaptado de ST Microelectronics (2000)

O circuito é construído com quatro chaves que são acionadas de forma alternada, S1 e S4 ou S2 e S3. Ou seja, para cada configuração das chaves o motor gira em um sentido. Deve-se observar que, as chaves S1 e S2 assim como as chaves S3 e S4 não podem ser ligadas ao mesmo tempo, pois podem gerar um curto circuito na fonte (ARDUINO, 2014).

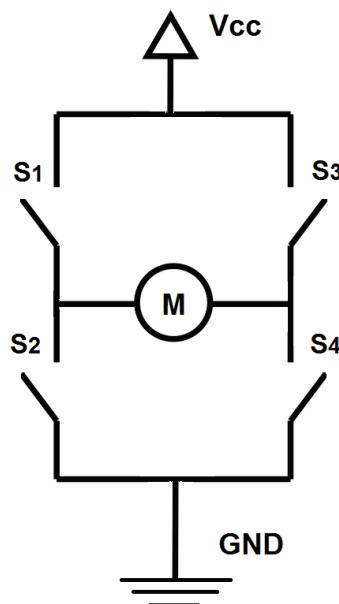


Figura 16 – Ponte H.

Fonte: Autoria própria

3.1.4 Microcontrolador Arduino

O Arduino é uma plataforma de computação embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de *hardware* e *software*. Em termos práticos, é um pequeno computador que se pode programar para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele (MCROBERTS, 2011).

Como pode ser visto na Figura 17, o Arduino Uno é uma placa baseada no microcontrolador ATmega328. Possui 14 pinos de entrada/saída digital, destes, 6 podem ser usados como saídas PWM e 6 entradas analógicas. Possui também cristal oscilador de 16MHz, conexão USB, entrada de alimentação, conexão ICSP e botão de reset (UNO, 2014).

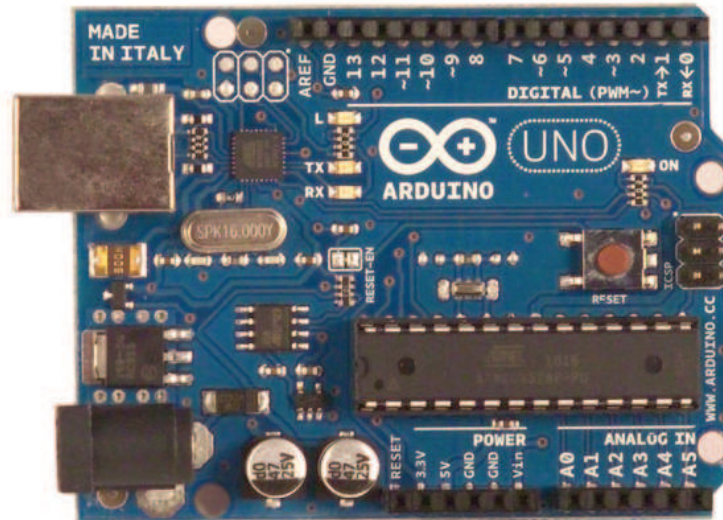


Figura 17 – Arduino Uno.

Fonte: McRoberts (2011)

O Arduino pode ser utilizado para desenvolver objetos interativos independentes, ou pode ser conectado a um computador, a uma rede, ou até mesmo à Internet para recuperar e enviar dados e atuar sobre eles (MCROBERTS, 2011).

Cada um dos 14 pinos digitais do Uno podem ser utilizados como uma entrada ou uma saída utilizando-se as funções **pinMode()**, **digitalWrite()**, e **digitalRead()**. Eles operam a 5V, com correntes de, no máximo, 40mA (UNO, 2014).

Para programar o Arduino, utiliza-se o IDE do Arduino, no qual se escreve o código em linguagem própria do Arduino (baseada na linguagem C) (MCROBERTS, 2011).

3.1.5 Sensor Pt-100

As termoresistências são sensores de temperatura usados amplamente nos processos industriais e laboratoriais, por suas condições de alta estabilidade, repetibilidade, resistência à contaminação, menor influência de ruídos e altíssima precisão de leitura (ALUTAL, 2014).

O sensor PT-100, Figura 18, é um tipo de termorresistência que mede a temperatura pela correlação da sua resistência elétrica, ou seja, apresenta o aumento da resistência com o aumento da temperatura. A maioria destes sensores é feita a partir de uma espiral de fio fino montada num suporte cerâmico ou de vidro. Possuem natureza frágil e necessitam ser instalados em bainhas protetoras (ECIL, 2014).

O material mais utilizado é a platina, pois o mesmo apresenta uma ampla escala de temperatura, alta resistividade permitindo maior sensibilidade, alto coeficiente de variação de resistência com a temperatura, uma boa linearidade resistência por temperatura e também a rigidez e dutibilidade para ser transformado em fios finos (ALUTAL, 2014).

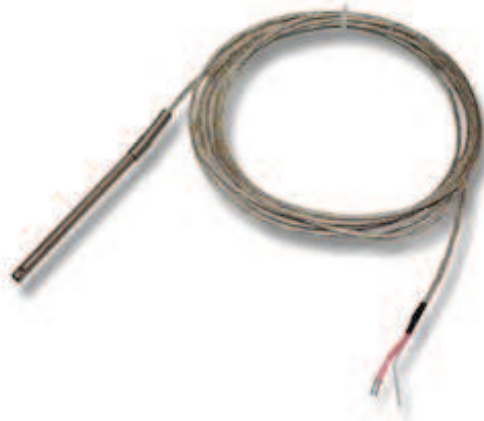


Figura 18 – PT-100.

Fonte: Ecil (2014)

Segundo Alutal (2014), de uma forma aproximada, a variação da resistência elétrica de um metal em função da temperatura pode ser representada pela Equação (34), onde $R(t)$ é a resistência elétrica, R_0 a resistência elétrica à temperatura de 0°C , α o coeficiente de variação da resistência elétrica em função da temperatura, T temperatura ($^{\circ}\text{C}$).

$$R(t) = R_0(1 + \alpha T) \quad (34)$$

Um estudo mais detalhado mostra que o coeficiente α varia em função da temperatura. Dentre os metais, mostram-se mais adequados na termometria de resistência, Cobre, Níquel, Liga de Rh99,5% x Fe0,5% e Platina.

As termoresistências são normalmente ligadas a um circuito de medição tipo Ponte de Wheatstone, sendo que o circuito encontra-se balanceado quando é respeitada a relação $R_4.R_2 = R_3.R_1$ (EDTEC, 2014), como mostra a Figura 19.

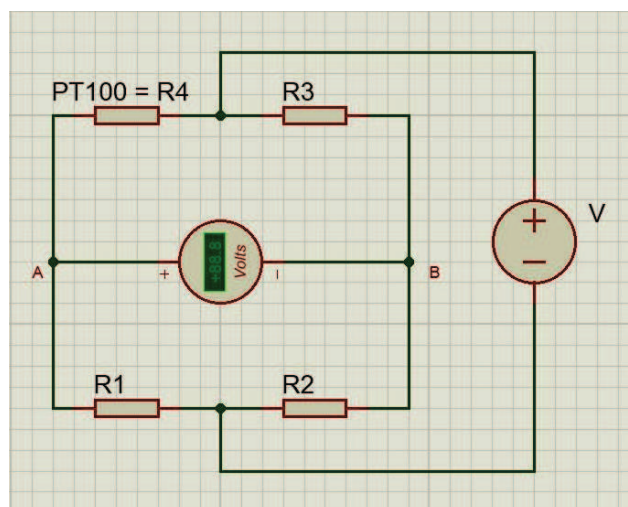


Figura 19 – Ponte de Wheatstone.

Fonte: Autoria própria

Como se vê na Figura 20, dois condutores de resistência relativamente baixa RL1 e RL2 são usados para ligar o sensor PT100 (R4) à ponte do instrumento de medição. Nesta disposição, a resistência R4 compreende a resistência do PT100 e mais a resistência dos condutores RL1 e RL2. Isto significa que se os fios RL1 e RL2 forem de baixa resistência, pode-se aumentar apreciavelmente a resistência do sensor, resultando em erro na leitura da temperatura. O método de ligação a dois fios, somente deve ser usado quando o sensor estiver à uma distância de aproximadamente 3 m e sempre que a temperatura ambiente ao longo dos fios de ligação variar, a leitura de temperatura do medidor introduzirá um erro, devido a variação da resistência de linha (EDTEC, 2014).

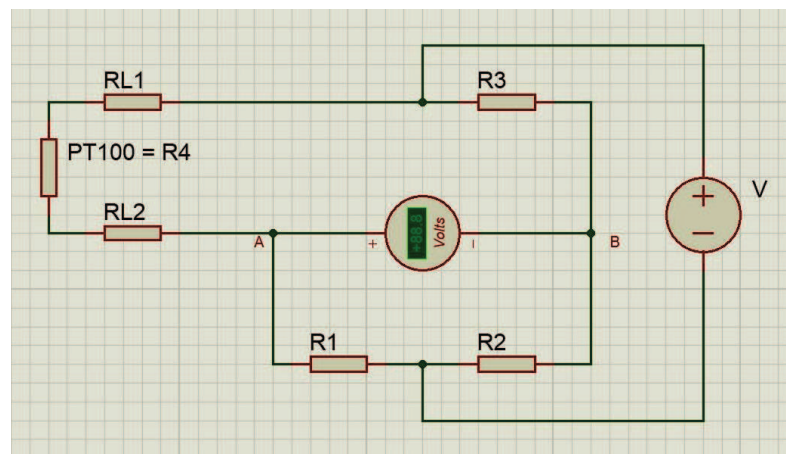


Figura 20 – Método de ligação a 2 fios.

Fonte: Autoria própria

3.1.6 Acionamento da resistência de aquecimento

Os tiristores SCR e TRIAC funcionam como chaves que podem obter disparo controlado, podendo ser usados para o controle de fase de tensão alternada e deixam de conduzir no instante que a corrente cai à zero. O TRIAC conduz corrente em ambos os sentidos, ou seja, o controle de fase pode ser feito tanto no semiciclo positivo quanto no semiciclo negativo (MARTINS et al., 2010). É empregado por exemplo nos reguladores de intensidade luminosa, no controle da velocidade de motores e no controle de resistências de aquecimento.

Foi implementado neste trabalho um circuito detector de passagem por zero utilizando um optoacoplador 4N35, para informar ao Arduino quando a tensão da rede elétrica passa por 0 V, Figura 21. A partir desse momento, calcula-se o tempo e ângulo de disparo do TRIAC, de maneira a ajustar o nível de potência entregue a resistência elétrica necessário para o controle da temperatura desejada.

Caso os disparos venham a ocorrer em qualquer outro ponto além do ponto zero da senóide, como mostra a Figura 22, a tensão liberada não mais será a tensão máxima, e sim uma tensão inversamente proporcional ao tempo de atraso do disparo do TRIAC (ALMEIDA, 2007). De acordo com Malvino (2008), na Figura 22, exemplo para disparo a 90° tem-se a curva tensão alternada da rede versus ângulo de disparo, onde a parte sombreada de cada semiciclo mostra quando o TRIAC está conduzindo. Controlando estes tempos de disparo pode-se controlar a tensão aplicada à carga.

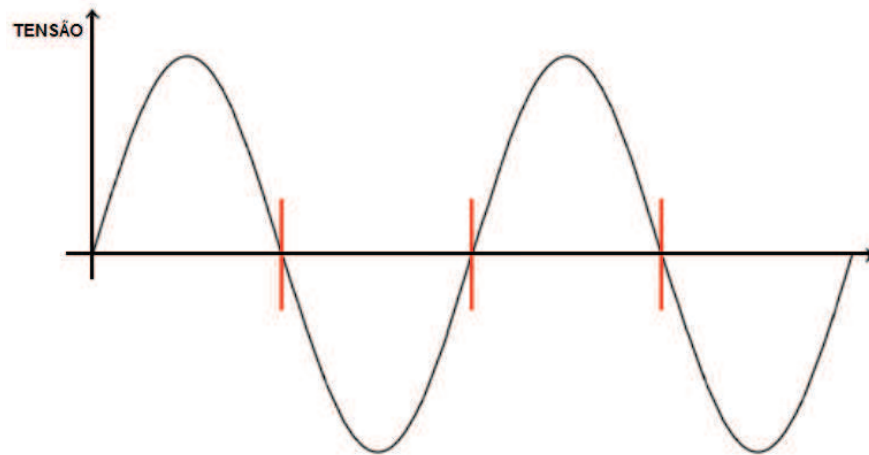


Figura 21 – Passagem por zero rede elétrica.
Fonte: Autoria própria

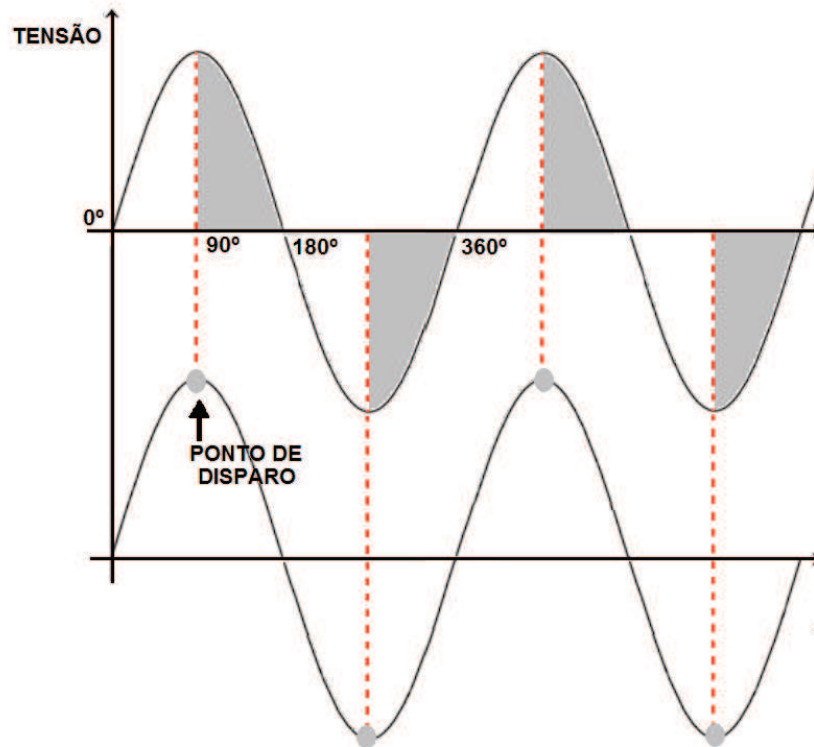


Figura 22 – Tensão e Ângulo de disparo.
Fonte: Autoria própria

Na Figura 23, mostra-se o gráfico de verificação do detector de passagem por zero, onde é possível perceber a tensão de referência em sincronia com a tensão da rede.

Com isto, é possível ter um completo controle sobre a carga, caso o disparo esteja sincronizado com a rede. Portanto, o ângulo de disparo do TRIAC varia de acordo com a posição inicial do pulso, e sendo feito tanto nos semiciclos positivos quanto nos semiciclos negativos da rede elétrica (MARTINS et al., 2010).

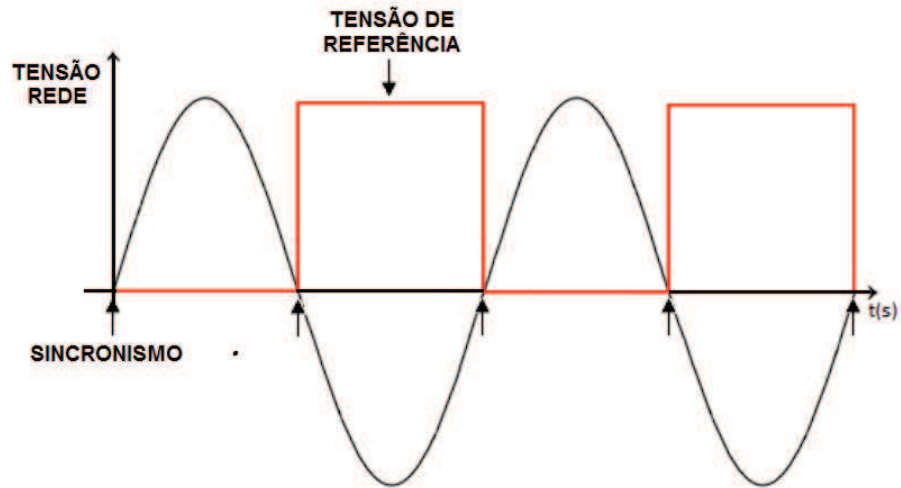


Figura 23 – Referência para a passagem por zero.

Fonte: Autoria própria

Como é possível variar a tensão na carga com a alteração do ângulo de disparo, pode-se perceber na Figura 24, que quanto menor o ângulo de disparo, maior a potência entregue a resistência elétrica, e conseqüentemente, maior a temperatura fornecida ao sistema (MARTINS et al., 2010).

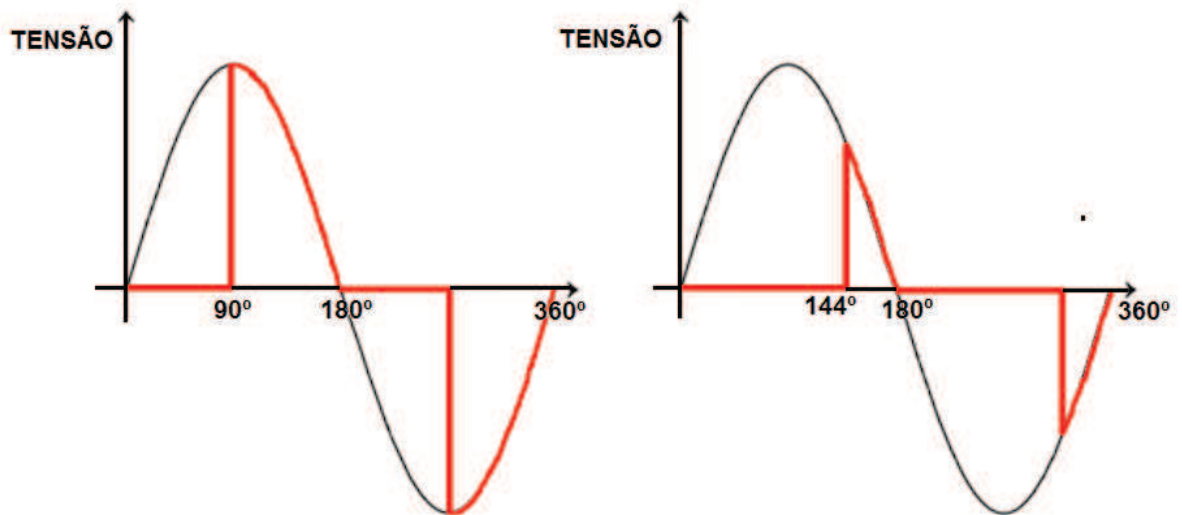


Figura 24 – Formas de onda de tensão na carga.

Fonte: Autoria própria

Utilizou-se neste trabalho o optoacoplador 4N35 para o circuito detector de passagem por zero, um MOC3020 e TIC 226D para o acionamento da resistência de aquecimento, como mostra Figura 25 e 26 respectivamente.

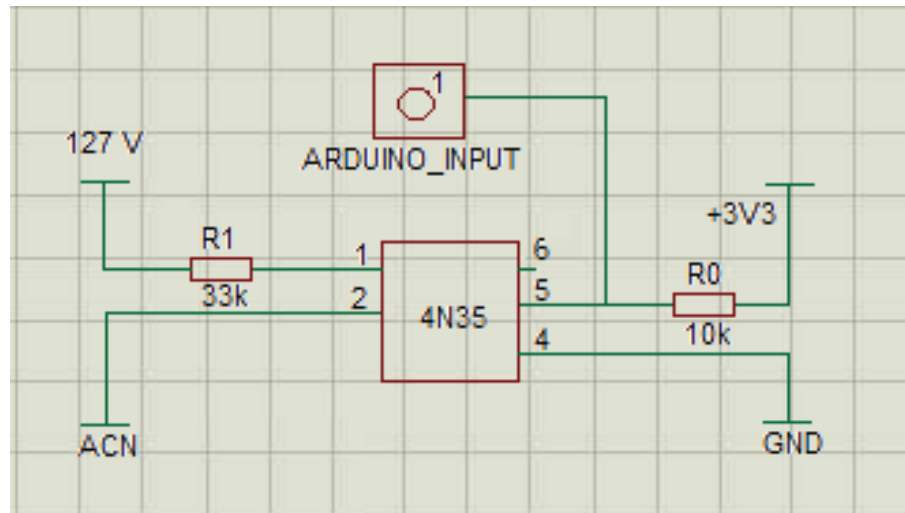


Figura 25 – Circuito detector de passagem por zero.

Fonte: Autoria própria

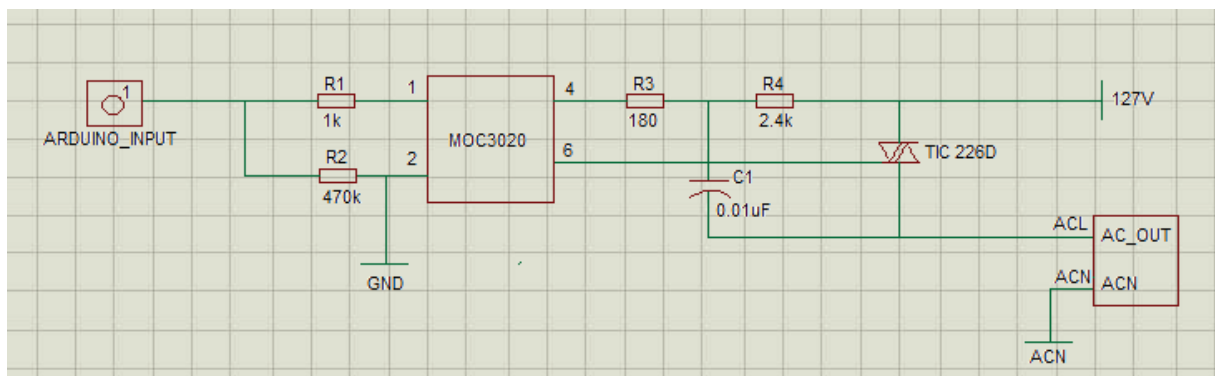


Figura 26 – Circuito de acionamento da resistência de aquecimento.

Fonte: Autoria própria

3.2 METODOLOGIA

Nesta seção apresenta-se o diagrama da metodologia dos sistema de nível e do sistema de temperatura, seguido da identificação do sistema e explicação do método de controle utilizado.

Como mencionado anteriormente, utilizou-se o microcontrolador Arduino para o processamento de sinais de ambos os sistemas, leitura dos sensores e converção dos sinais analógicos para digitais e sinais digitais para PWM. As estratégias de controle implementadas para cada sistema foi realizadas no software MATLAB e com comunicação por USB.

A Figura 27 apresenta o diagrama de blocos da metodologia de controle do sistema de nível, onde este possui um sensor ultrassônico HC-SR04, que envia os dados do nível para o Arduino 1. O Arduino 1, por sua vez, recebe a informação do nível e converte em um sinal PWM. Em seguida, para transformar tal sinal em um nível de tensão analógico, utilizamos um filtro RC (resistor-capacitor) em série. O Arduino 2 recebe do filtro RC um sinal de tensão proporcional ao nível do líquido. No Arduino 2 conectado diretamente com o MATLAB realiza-

se a ação do controlador e envia sinais PWM para o acionamento da bomba de corrente contínua do reservatório.

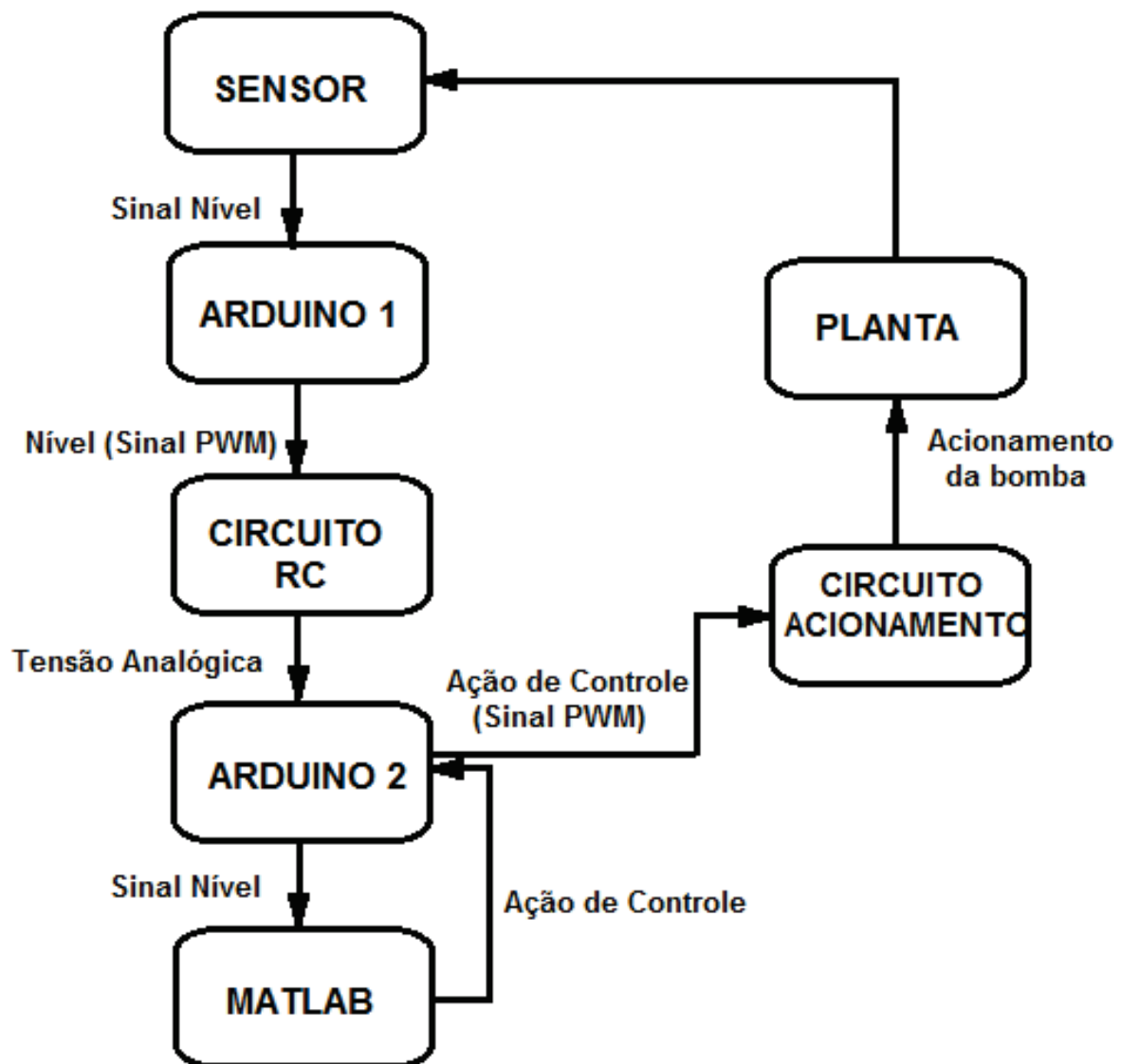


Figura 27 – Diagrama do controle do Sistema de Nível.

Fonte: Autoria própria

De forma similar, na Figura 28, tem-se o diagrama de blocos do controle do sistema de temperatura. Utilizou-se o sensor PT100 que envia diretamente seu sinal ao Arduino 2. O Arduino 2 conectado ao MATLAB realiza a ação de controle. Dessa forma ele envia o sinal de controle ao circuito RC que converte em sinal analógico. O Arduino 3 foi adicionado ao sistema devido a necessidade de sincronismo da rede com o disparo do TRIAC. Portanto, o Arduino 3 sincronizado com a rede recebe o sinal analógico da ação de controle e converte para sinal PWM necessário para o acionamento do circuito de aquecimento.

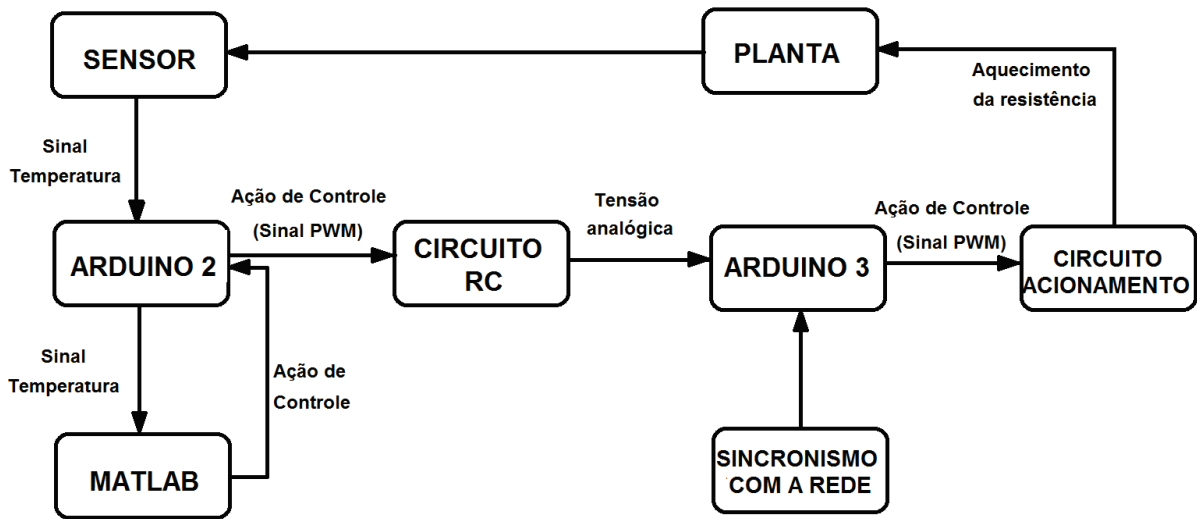


Figura 28 – Diagrama do controle do sistema de temperatura.

Fonte: Autoria própria

3.2.1 Controle Clássico para o Sistema de Nível

Primeiramente, para a implementação da técnica de controle clássico no sistema de nível, foi realizado o ensaio em malha aberta, Figura 29. Para o ensaio, utilizou-se um valor fixo de 1,667V aplicado a bomba do reservatório.

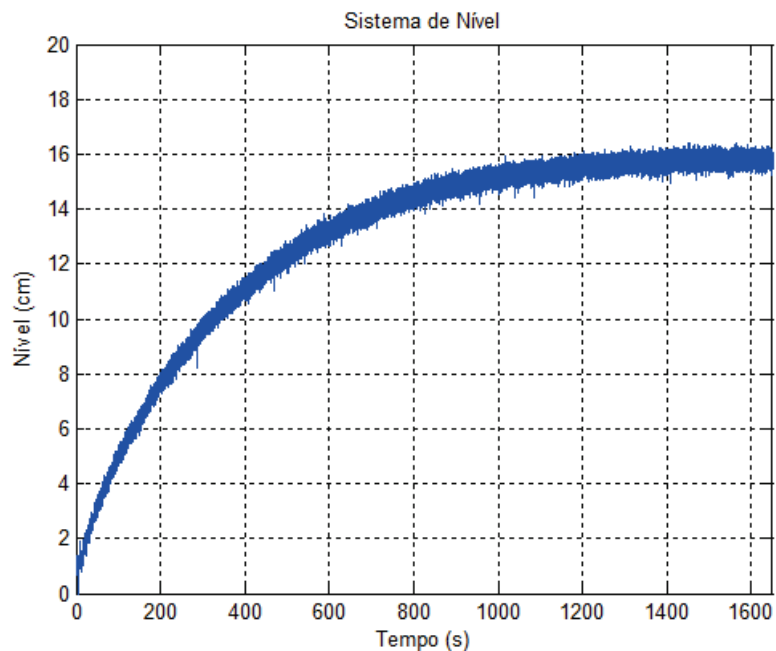


Figura 29 – Ensaio em malha aberta para o sistema de nível.

Fonte: Autoria própria

Este sistema pode ser considerado de primeira ordem sem atraso de transporte, portanto, para determinar os valores de K e τ utiliza-se o método de identificação do sistema de

primeira ordem sem atraso de transporte apresentado na Seção 2.2.1. Na Figura 30, apresenta-se o ensaio em malha aberta do sistema de nível e a função de transferência identificada em malha aberta, conforme Equação (35).

$$G(s) = \frac{8.853}{320s + 1} \quad (35)$$

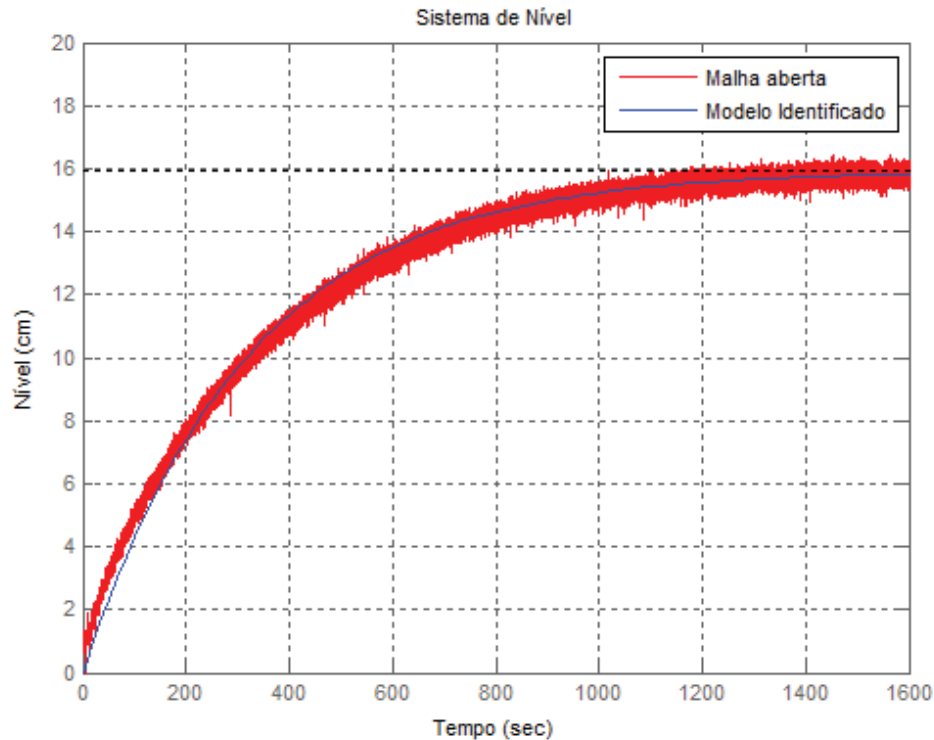


Figura 30 – Ensaio em malha aberta e função de transferência identificada.

Fonte: Autoria própria

Neste caso, será implementado um controlador PI suficiente para fazer com que a resposta em malha fechada a uma entrada de 1.667V aplicada tenha erro estacionário nulo. Assim usando-se as Equações (10) e (11) obtemos $K_p = 0,036$, $K_i = 320$, e a função de transferência do controlador, Equação (36).

$$G_c(s) = K_p + \frac{K_i}{s} = 0,036 + \frac{320}{s} \quad (36)$$

3.2.2 Controle Moderno para o Sistema de Nível

Para comparação com a técnica de controle clássico aplicado ao sistema de nível, optou-se por implementar a técnica de controle moderno por realimentação de estados utilizando o método de alocação de polos.

Supondo que o sistema seja linear e de segunda ordem, então o modelo matemático da saída é dado pelas Equações (37) e (38) para o nível e velocidade respectivamente.

Essa velocidade é obtida pela derivada do sinal original do nível em malha aberta, Figura 31. Para a realização dos ensaios de controle, utilizou-se um filtro passa baixa *Butter-*

worth de ordem 6 e frequência de corte de 0.01 aplicado ao sinal do nível. Dessa forma, com a redução dos ruídos realizou-se a derivada do sinal obtendo-se assim a velocidade.

$$N_{ivel} = C_1 e^{C_2 t} + C_3 e^{C_4 t} + C_5 \quad (37)$$

$$V_{elocidade} = C_6 e^{C_2 t} + C_7 e^{C_4 t} + C_8 \quad (38)$$

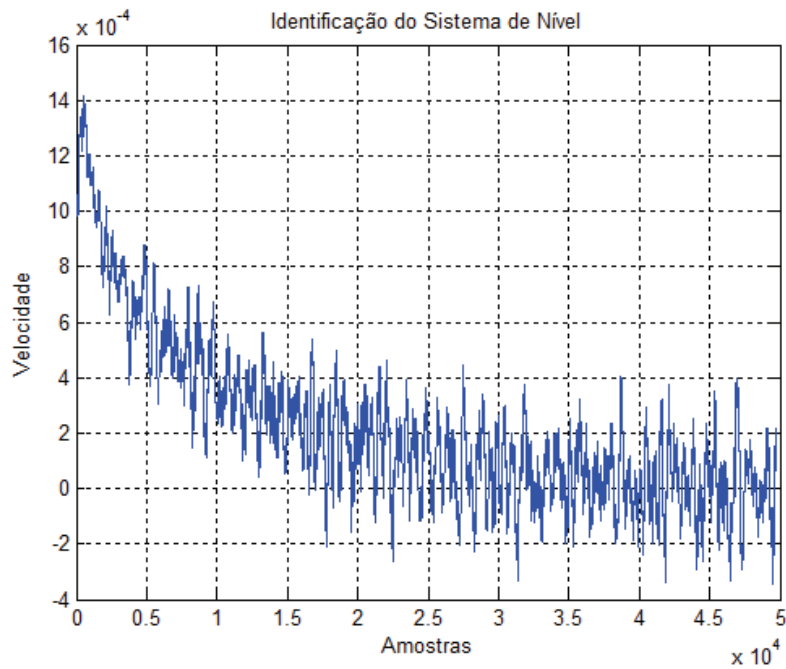


Figura 31 – Velocidade do sistema de nível obtida pela derivada.

Fonte: Autoria própria

Usando o comando `LSQNONLIN` do MATLAB, calculou-se as constantes para melhor ajustar o modelo matemático às funções obtidas na prática, sendo $C_2 = -0,00009217$ e $C_4 = -8,0014$ os polos em malha aberta. Com o modelo matemático da dinâmica dos estados, foram determinadas as matrizes abaixo.

$$A = 10^5 * \begin{bmatrix} -0.0002 & -3.0424 \\ -0.0000 & -0.0013 \end{bmatrix},$$

$$B = \begin{bmatrix} 209.7767 \\ 0.0883 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Com base nas matrizes obtidas, pode-se resolver o problema de alocação de polos com o *software* MATLAB. Problemas de alocação de polos podem ser facilmente resolvidos com o MATLAB, o qual possui dois comandos (`acker` e `place`) para o cálculo da matriz de ganho \mathbf{K} de realimentação (OGATA, 2010).

Nas Figuras 32 e 33, tem-se a identificação do sistema a partir dos valores das matrizes para o nível e para a velocidade do sistema, respectivamente.

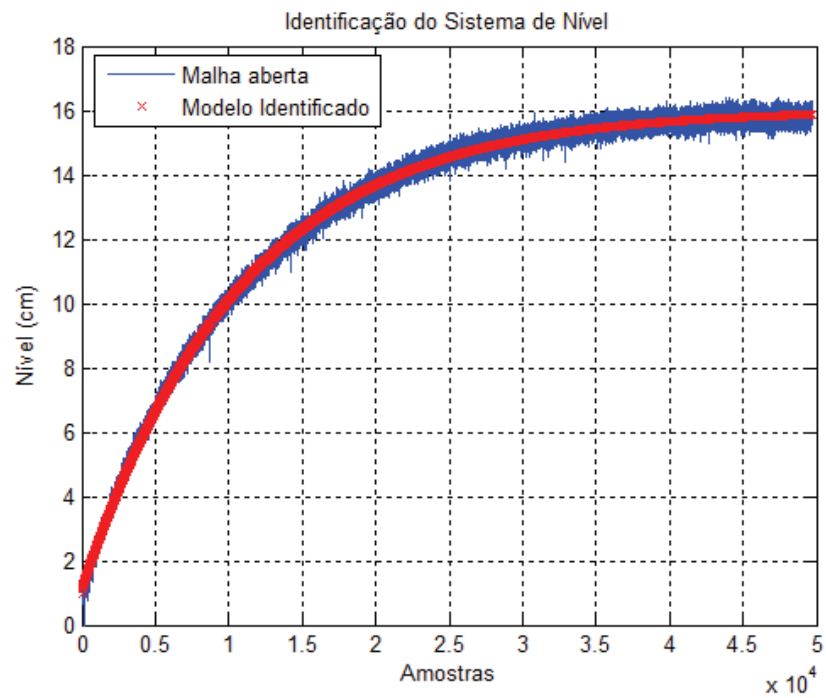


Figura 32 – Resposta em malha aberta para o sistema de nível.

Fonte: Autoria própria

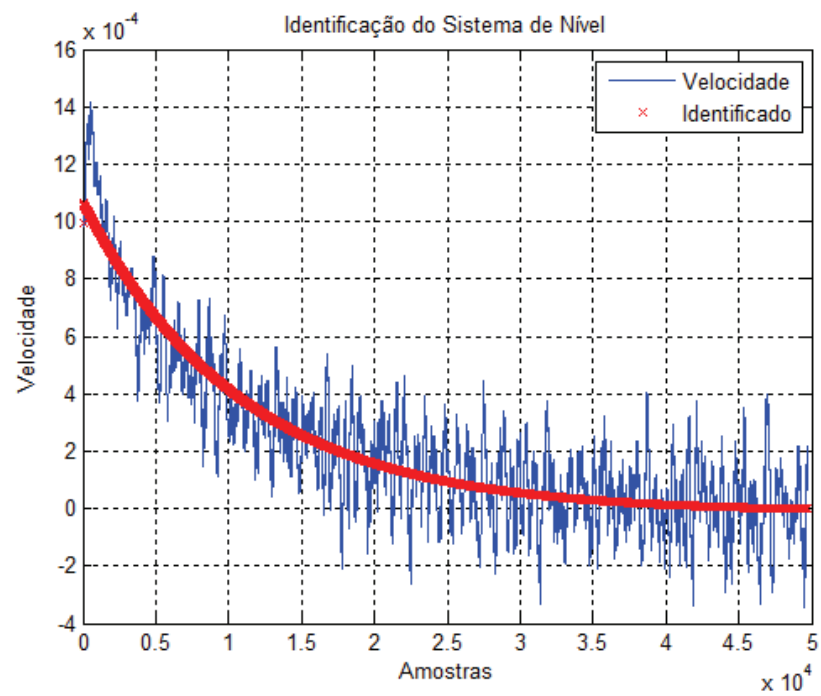


Figura 33 – Resposta em malha aberta para variável velocidade.

Fonte: Autoria própria

Ainda segundo Ogata (2010), se o sistema envolver múltiplas entradas, a matriz \mathbf{K} não será única e tem-se graus de liberdade adicionais para escolher \mathbf{K} . Dessa forma, a alocação de polos baseada nessa abordagem é denominada alocação robusta de polos, então utiliza-se o comando `place` no MATLAB.

Dessa forma, para utilizar o comando `place`, fornecem-se polos desejados de malha fechada, os quais surgiram a partir da referência dos polos de malha aberta das Equações (37) e (38), variáveis C_2 e C_4 .

Não é fácil decidir aonde alocar os polos de um determinado sistema que se deseja controlar, é importante observar que polos estáveis muito distantes do eixo imaginário geram as seguintes implicações:

- O sistema em malha fechada responde mais rapidamente;
- Os ganhos de realimentação tornam-se elevados;
- O controlador deve gerar um sinal de grande amplitude para controlar o sistema;

Deve-se considerar também que, quanto mais distante os polos do sistema em malha fechada estiverem dos polos de malha aberta, maior devem ser os módulos dos elementos do ganho de realimentação \mathbf{K} . Dessa forma, a análise da melhor alocação dos polos será vista com mais detalhes na Seção 4.1.2.

3.2.3 Controle Clássico para o Sistema de Temperatura

A mesma análise utilizada na técnica de controle clássico para o sistema de nível, aplica-se ao sistema de temperatura. Inicialmente é realizado o controle em malha aberta, como mostra a Figura 34, e realizada a identificação do sistema com a obtenção da função de transferência.

Podendo ser considerado também um sistema de primeira ordem, obteve-se assim, a função de transferência em malha aberta, Equação (38).

$$G(s) = \frac{272.57}{1171s + 1} \quad (39)$$

Na Figura 35, apresenta-se o ensaio em malha aberta do sistema de temperatura e a função de transferência identificada, onde os parâmetros de K e τ foram determinados utilizando o método de identificação do sistema de primeira ordem sem atraso de transporte apresentado na Seção 2.2.1.

Com isso, da mesma forma que para o sistema de nível, um controlador PI é suficiente para o controle do sistema, com valores de $K_p = 0.001586$ e $K_i = 1171$ definidos a partir das Equações (10) e (11), que serão utilizados como parâmetros iniciais para tal controle. Dessa forma, tem-se pela Equação (40) a função de transferência do controlador.

$$G_c(s) = K_p + \frac{K_i}{s} = 0.001586 + \frac{1171}{s} \quad (40)$$

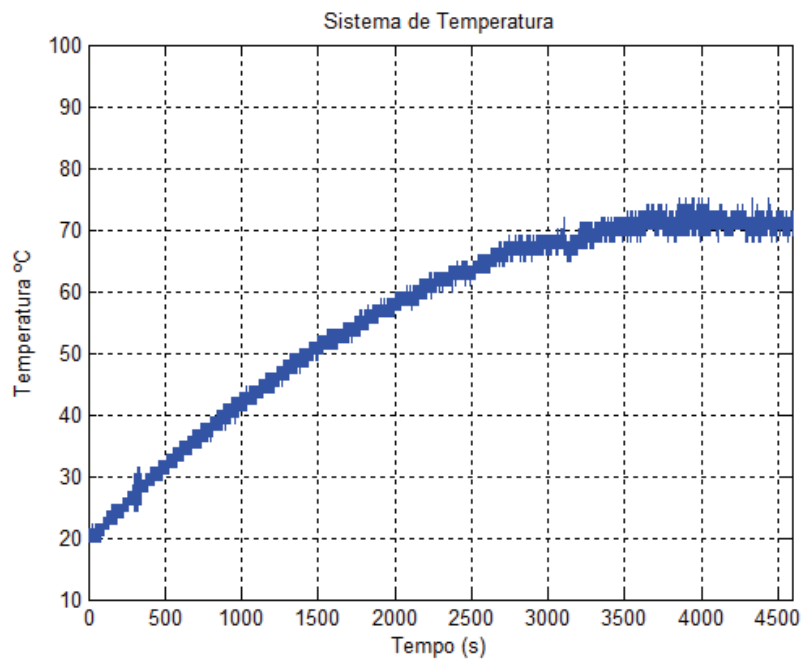


Figura 34 – Ensaio em malha aberta para o sistema de temperatura.

Fonte: Autoria própria

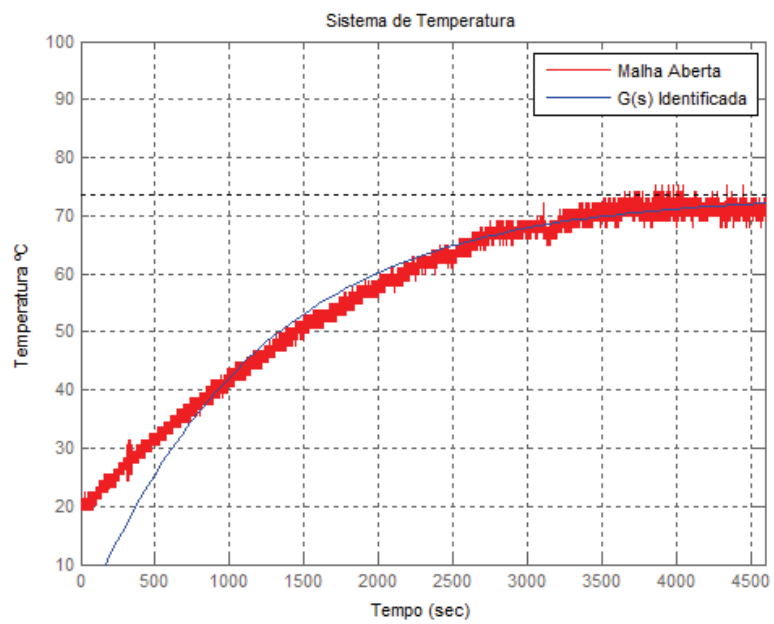


Figura 35 – Sistema de Temperatura em malha aberta e identificada.

Fonte: Autoria própria

4 RESULTADOS E DISCUSSÕES

Neste capítulo, apresentam-se os principais resultados obtidos para cada um dos sistemas e os respectivos tipos de controle aplicados. Em seguida, apresentam-se discussões comparativas entre os dois tipos de controle aplicados ao sistema de nível e, por fim, análise para comprovação do controle clássico aplicado ao sistema de temperatura.

4.1 RESULTADOS DO SISTEMA DE NÍVEL

4.1.1 Controle Clássico

Para a comprovação do funcionamento do controle clássico utilizando o controlador PI para o sistema de nível, os parâmetros utilizados foram $K_p = 0,036$ e $K_i = 320$ obtidos a partir da identificação do sistema em malha aberta. O Ensaio 1 foi realizado com apenas um *set point*, como mostra a Figura 36.

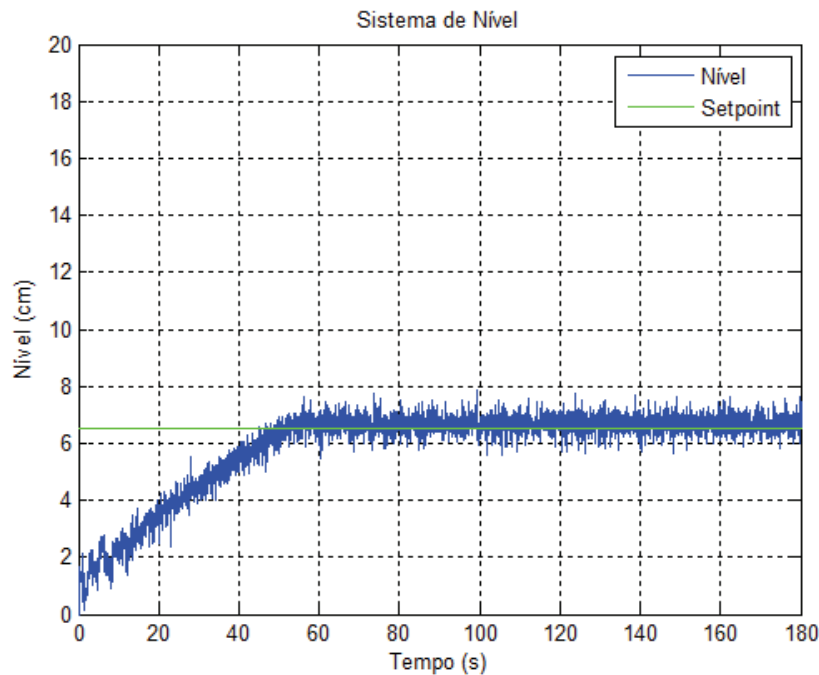


Figura 36 – Ensaio 1 - Controle PI para sistema de nível.

Fonte: Autoria própria

Embora os valores de K_p e K_i sintonizados são apenas sugestões iniciais para a utilização do controlador, com base na Figura 36, observa-se que o controle foi eficiente a partir desses parâmetros. O tempo para o sistema entrar em regime permanente foi em torno de 60 segundos.

Nota-se na Figura 37, que o Ensaio 2 - Controle PI foi realizado utilizando 3 *set points* diferentes, onde estes, foram determinados de modo a analisar se o controle se comporta de maneira eficiente para busca-los. O sistema entrou em regime permanente para o primeiro *set point* em torno de 140 segundos.

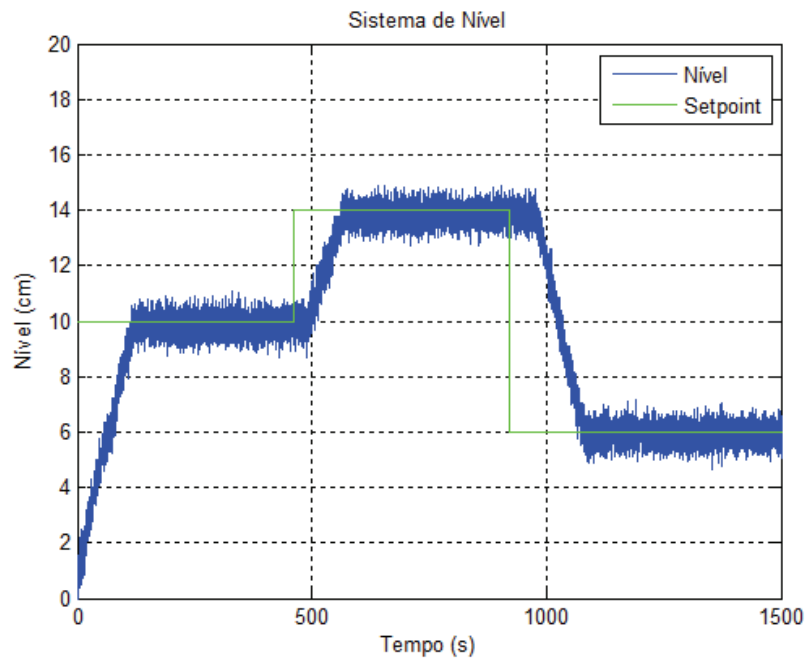


Figura 37 – Ensaio 2 - Controle PI para sistema de nível.

Fonte: Autoria própria

Ao alterar os *set points* em ordem decrescente, a resposta do controle do nível pode ser observada no Ensaio 3 - Controle PI, Figura 38. Da mesma maneira que o Ensaio 1 os parâmetros K_p e K_i se comportaram de forma precisa e eficiente para os Ensaio 2 e 3.

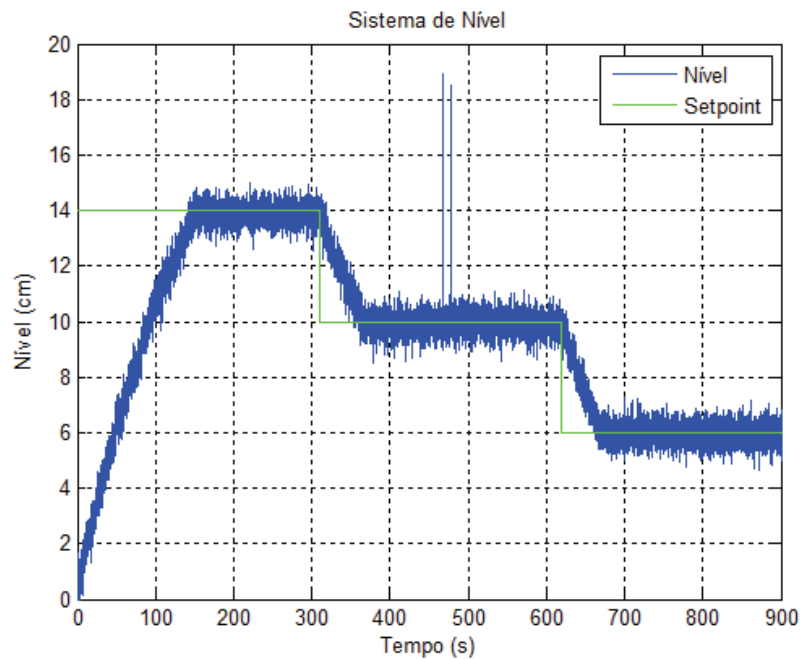


Figura 38 – Ensaio 3 - Controle PI para sistema de nível.

Fonte: Autoria própria

4.1.2 Controle Moderno

Conforme mencionado anteriormente, para a técnica de controle moderno por realimentação de estados utilizando o método de alocação de polos, foi necessário a utilização de um Integrador no sistema, Figura 9. Este bloco integrador adiciona ao sistema outra variável de estado aplicada ao sinal de controle. Levando isto em consideração, e os valores dos polos em malha aberta, constantes C_2 e C_4 , realiza-se três ensaios diferentes com intuito de encontrar o melhor resultado para o ganho de realimentação \mathbf{K} , Figuras 39, 40 e 41.

O Ensaio 1 - Alocação de polos, Figura 39, foi realizado com os seguintes valores desejados dos polos em malha fechada $J = \begin{bmatrix} -0.8 & -0.1 & -0.5 \end{bmatrix}$, utilizando o comando `place` obtemos os seguintes valores de ganho $K = \begin{bmatrix} 924.9 & 9631.7 & -40.7 \end{bmatrix}$. Os valores dos polos definidos da matriz \mathbf{J} surgiram a partir da referência dos polos em malha aberta. Nota-se neste ensaio que o sistema entrou em regime permanente em torno de 150 segundos.

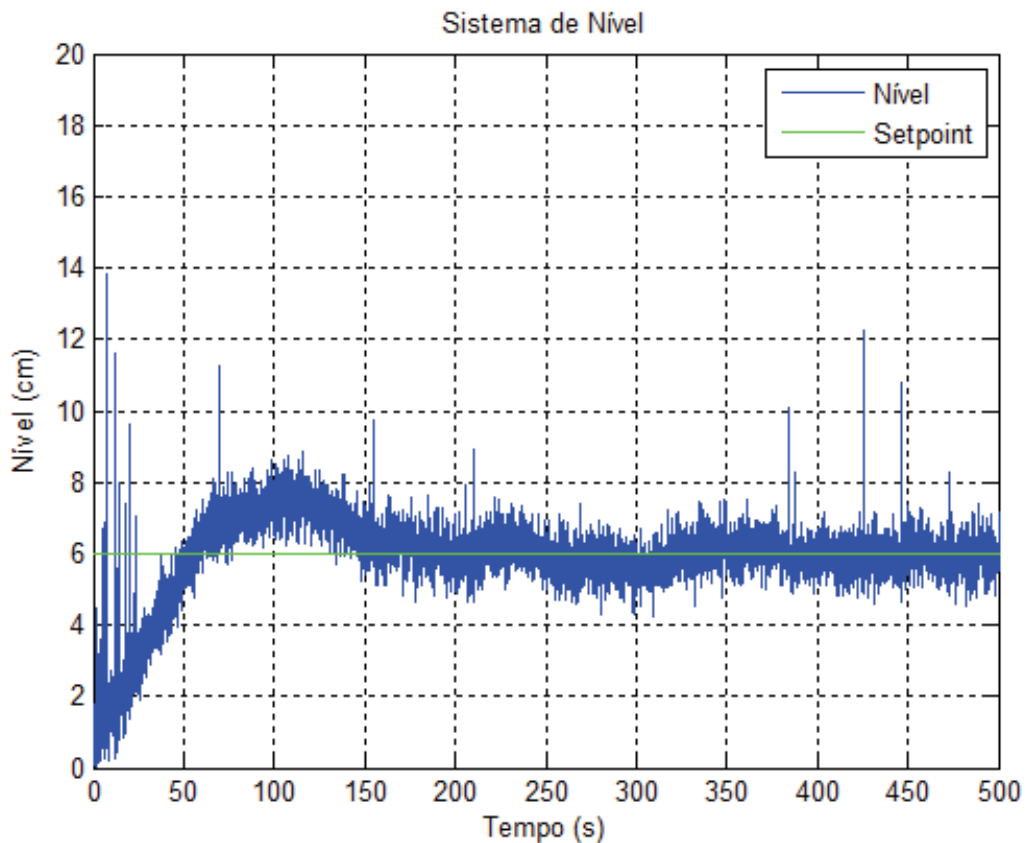


Figura 39 – Ensaio 1 - Alocação de polos.

Fonte: Autoria própria

Utilizou-se para o Ensaio 2 - Alocação de polos $J = \begin{bmatrix} -0.8 & -0.4 & -0.5 \end{bmatrix}$ e obteve-se $K = \begin{bmatrix} 2478 & 23885 & -163 \end{bmatrix}$ como ganho de realimentação, Figura 40.

E por fim, o Ensaio 3 - Alocação de polos, realizado com $J = \begin{bmatrix} -0.8 & -0.5 & -0.7 \end{bmatrix}$ obtem-se $K = \begin{bmatrix} 4031 & 38138 & -285 \end{bmatrix}$, Figura 41. Pode-se notar que há uma pequena diferença nos resultados dos Ensaio 2 e 3, onde ambos entraram em regime permanente a partir de 250 segundos.

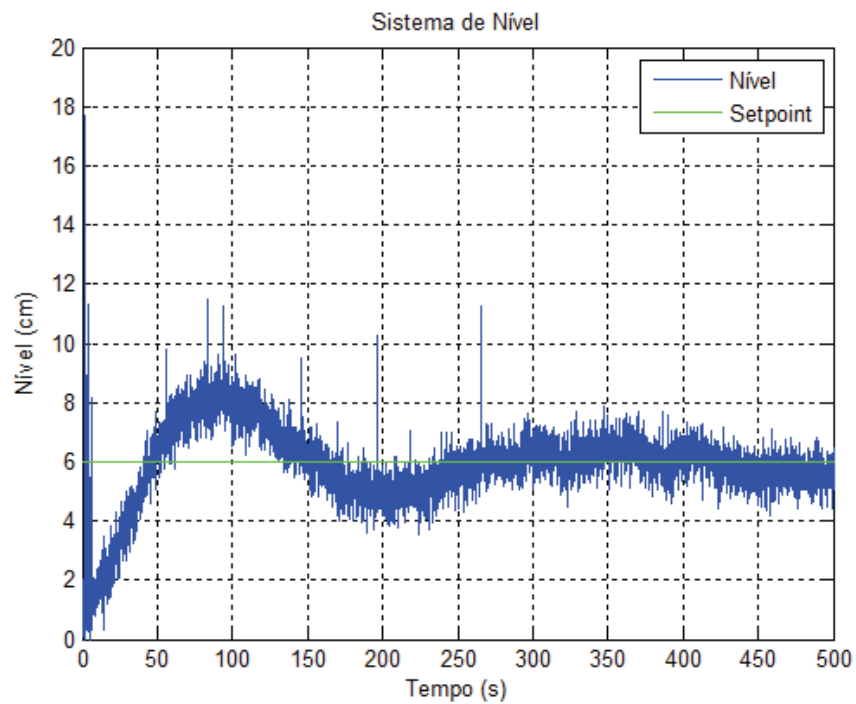


Figura 40 – Ensaio 2 - Alocação de polos.
Fonte: Autoria própria

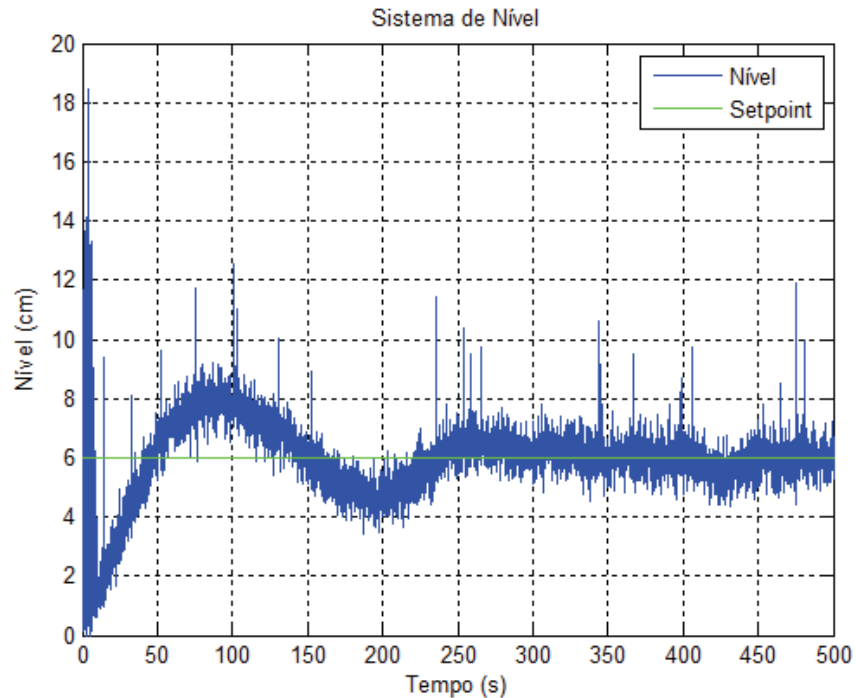


Figura 41 – Ensaio 3 - Alocação de polos.
Fonte: Autoria própria

Como já mencionado anteriormente, a decisão de onde alocar os pólos não é fácil, levando em consideração suas implicações e analisando as Figuras 39, 40 e 41, utiliza-se o

ganho de realimentação \mathbf{K} do Ensaio 1 - Alocação de polos, pois este mostrou ter um comportamento mais próximo ao desejado. Pode-se perceber na Figura 42, que o controle moderno possui um valor de *overshoot* de aproximadamente 2 centímetros, e comparado ao controle PI nota-se que o sistema entrou em regime permanente a partir de 150 segundos.

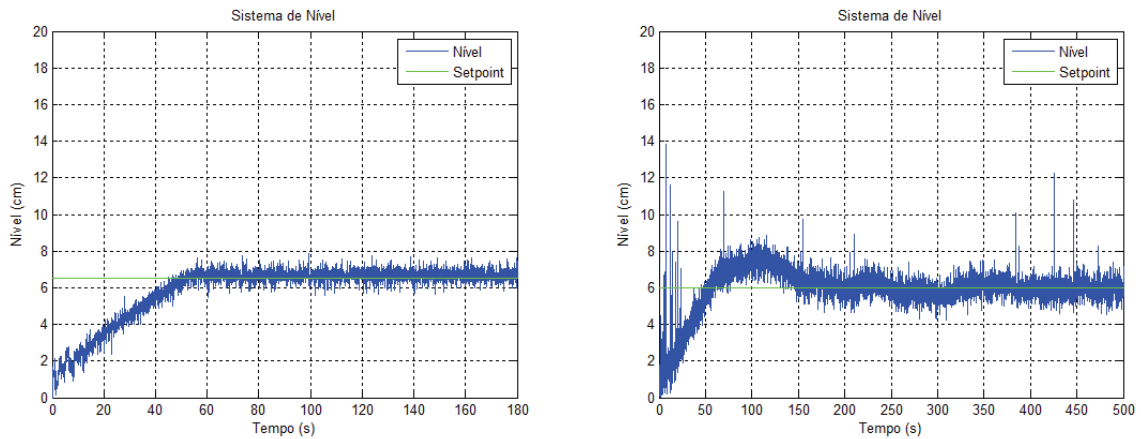


Figura 42 – Comparação: a) Ensaio 1 - Controle PI. b) Ensaio 1 - Alocação de polos.

Fonte: Autoria própria

Com os valores de \mathbf{K} definidos, realiza-se o Ensaio 1 - Controle por Espaço de Estados, Figura 43, utilizando três *set points* diferentes para efeitos comparativos, seguindo a mesma análise do Ensaio 2 - Controle PI, Figura 37.

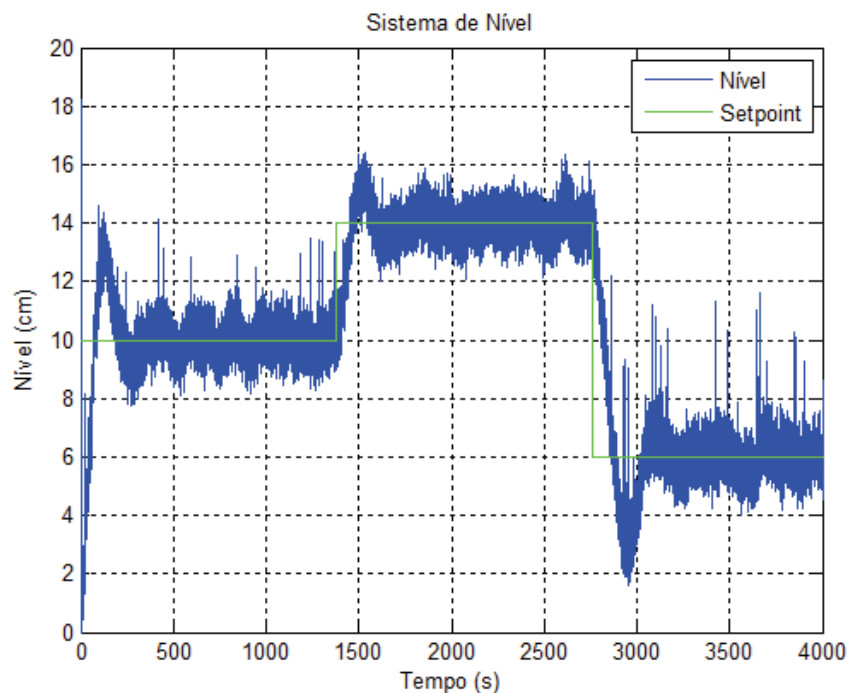


Figura 43 – Ensaio 1 - Controle por realimentação de estados.

Fonte: Autoria própria

Nota-se na Figura 44, que o Ensaio 2 - Controle PI mostra-se mais preciso, eficiente e com resposta mais rápida em relação ao Controle por realimentação de estados.

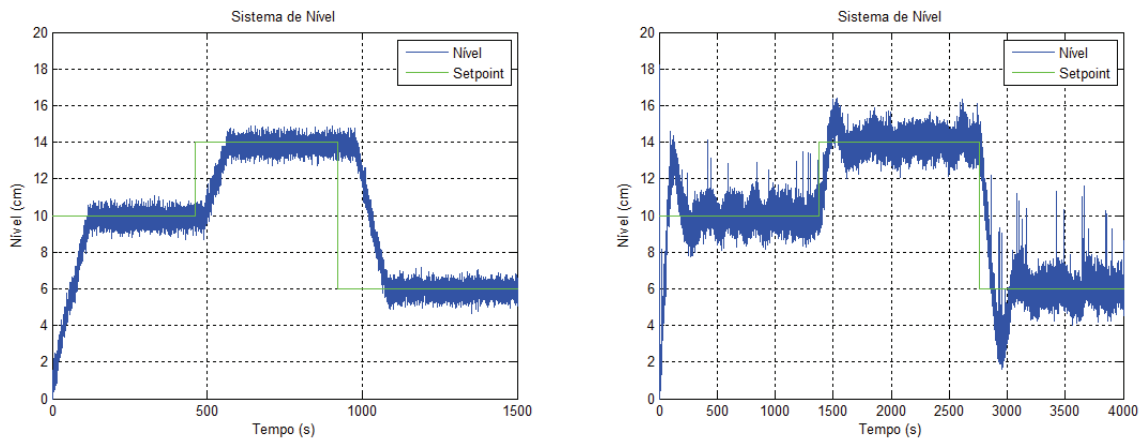


Figura 44 – Comparação: a) Ensaio 2 - Controle PI. b) Ensaio 1 - Controle por realimentação de estados.
Fonte: Autoria própria

A Figura 45, Ensaio 2 - Controle por realimentação de estados foi realizado de modo a comprovar seu funcionamento e comparação com o Ensaio 3 - Controle PI.

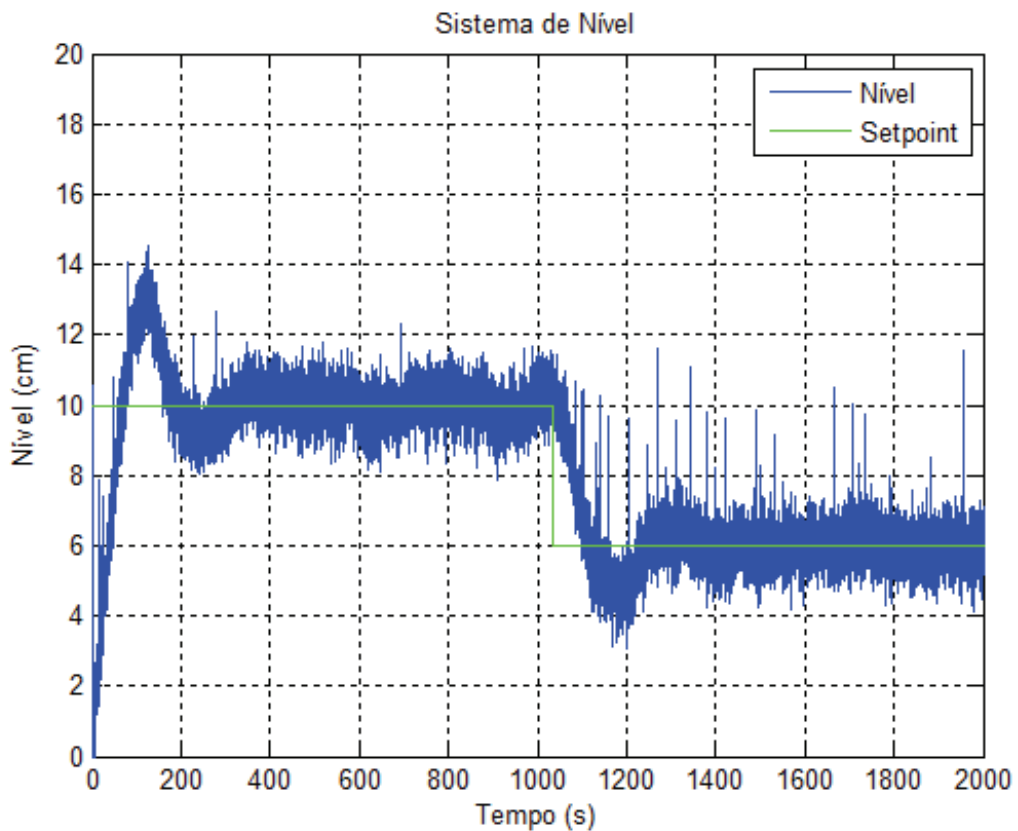


Figura 45 – Ensaio 2 - Controle por realimentação de estados.
Fonte: Autoria própria

Contudo, não foi possível a utilização de três *set points* diferentes para comparação com o Ensaio 3 - Controle PI, devido ao máximo *overshoot* do sistema do controle moderno e limitação do reservatório de líquido, pois, para o *set point* de 14 cm ocorreria um *overshoot* de aproximadamente 4 cm, com isso haveria falta de água no reservatório inferior limitando o reservatório superior alcançar 18 cm.

4.2 RESULTADOS DO SISTEMA DE TEMPERATURA

Os valores iniciais dos parâmetros de $K_p = 0.001586$ e $K_i = 1171$ foram definidos a partir da identificação do sistema em malha aberta. Embora esses valores sejam apenas o ponto de partida para o controle, pode-se analisar que não foi possível a utilização destes parâmetros devido ao sistema não responder a aplicação do controle.

Dessa forma, no Ensaio 1 - Temperatura utilizou-se os seguintes parâmetros: $K_p = 0.01586$ e $K_i = 10$, que foram definidos de forma manual. Por fim, inicia-se o ensaio com *set point* no valor de 30°C , Figura 46.

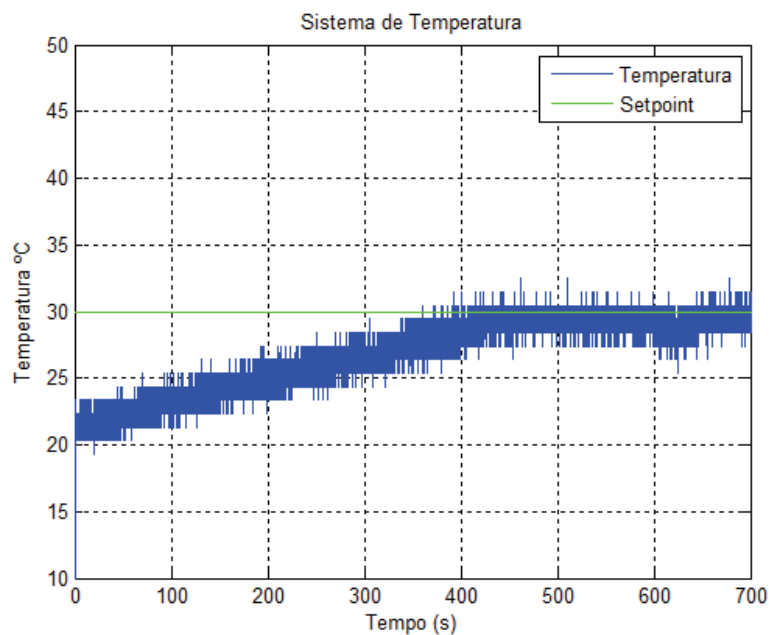


Figura 46 – Ensaio 1 - Temperatura.

Fonte: Autoria própria

Nota-se também no Ensaio 2 - Temperatura, Figura 47 que o sistema de controle clássico se mostra eficiente e busca de maneira adequada os dois *set points*, primeiro de 30°C e o segundo de 40°C .

E por fim, o Ensaio 3 - Temperatura foi realizado a fim de comprovar que ambos os sistemas funcionam de forma simultânea. Para desconsiderar efeitos multivariáveis realizou-se apenas o controle do sistema de temperatura a partir do momento em que o sistema de nível já estivesse estabilizado de modo a não danificar o aquecedor, Figura 48. Pode-se notar que houve uma grande interferência no sinal de temperatura devido ao sistema do nível.

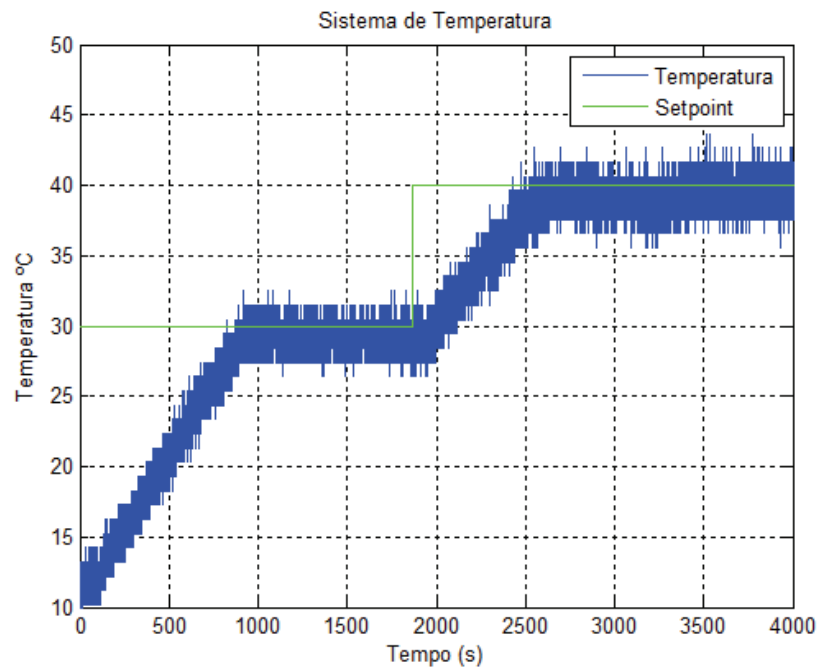


Figura 47 – Ensaio 2 - Temperatura.

Fonte: Autoria própria

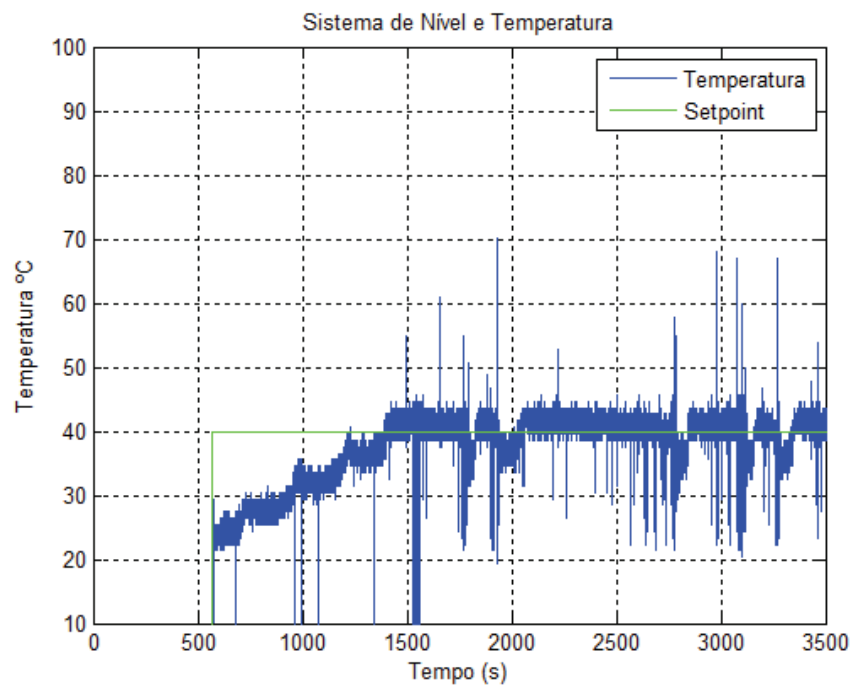


Figura 48 – Ensaio 3 - Temperatura.

Fonte: Autoria própria

5 CONCLUSÕES

Neste capítulo, apresentam-se as conclusões dos resultados obtidos neste trabalho, sugerindo melhorias para os próximos trabalhos.

5.1 CONSIDERAÇÕES FINAIS

Foi apresentado, neste trabalho, um estudo sobre técnicas de controle que podem ser aplicadas tanto para o sistema de nível quanto para o sistema de temperatura. O reservatório de líquido utilizado para este trabalho foi extraído do Módulo de Nível e Temperatura 2325 do fabricante Datapool.

Desenvolveu-se uma interface de aquisição de dados utilizando o microcontrolador Arduino, e sistema de controle com *software* MATLAB, o qual foi usado como uma ferramenta de controle e supervisão dos sistemas implementados na planta.

Após coletados os dados do processo utilizando o Arduino, foram aplicados alguns métodos de identificação de sistemas de primeira ordem tanto para o sistema de nível quanto para o sistema de temperatura. Com os modelos matemáticos obtidos a partir da identificação, pode-se aplicar técnicas de controle clássico e controle moderno. Os métodos de identificação apresentados neste trabalho são métodos práticos e de aceitação em aplicações industriais, os quais exigem pouco esforço computacional. Observa-se que os modelos matemáticos identificados são modelos aproximados do sistema real.

Apresentou-se um sistema para o controle de malha fechada de temperatura de líquidos, e para sua implementação foi utilizado o sensor de temperatura PT100. Também foi desenvolvido um circuito de aquisição dos sinais utilizando um Arduino, onde este envia o sinal para o MATLAB que é responsável pelo controle, e outro circuito utilizando um TRIAC para o acionamento da resistência de aquecimento.

Realizou-se três ensaios diferentes para o sistema de temperatura, onde foi implementado para ambos o Controle PI. Dessa forma, os Ensaios 1 e 2 obtiveram valores satisfatórios a partir da identificação do sistema. Para o Ensaio 3, realizou-se o controle da temperatura para o sistema completo, onde o nível do reservatório de líquidos foi mantido constante e o suficiente para não danificar o aquecedor. Essa estratégia foi realizada de modo a poder desconsiderar efeitos multivariáveis. Nota-se então, que este sinal de controle mostrou-se eficiente mesmo com a interferência do sistema de nível, porém com muito ruído.

Com base nos resultados do sistema de nível, conclui-se a partir dos ensaios realizados utilizando o controle clássico e o controle moderno que os resultados mais satisfatórios foram utilizando o controlador PI. Pode-se notar que no sistema de controle PI, todos os ensaios chegaram ao regime permanente mais rápido que o controle por realimentação de estados, e com respostas mais precisas referentes aos *set points*. Isto pode ser justificado pelo fato de que houve um significativo valor de *overshoot* na resposta do sistema controlado por realimentação de estados. Para a melhoria desse sistema, seria necessário um estudo mais específico referente à alocação de polos em malha fechada, o que traria uma significativa melhora na resposta do sistema.

Os objetivos deste trabalho para o sistema de nível e para o sistema de temperatura foram portanto atingidos, visto que os sistemas que foram implementados apresentaram

respostas satisfatórias. Ainda assim, existem possibilidades de aprimoramento deste projeto, visando um uso mais eficaz do sistema.

5.2 SUGESTÕES PARA TRABALHOS FUTUROS

Para trabalhos futuros, sugere-se aqui um estudo da viabilidade da utilização de outros tipos de sensores que melhor venham a se adequar às características de funcionamento do sistema ou até mesmo estudar formas de calibrá-los. Na calibração dos sensores, tanto para o sistema de nível quanto para o sistema de temperatura, podem ser aplicadas técnicas de filtragem digital para diminuir os ruídos.

A implementação do controle moderno por realimentação de estados utilizando o método por alocação de polos, gerou um significativo valor de *overshoot* devido a alocação dos polos em malha aberta e grande variação do nível. Em trabalhos futuros sugere-se estudar e aplicar técnicas de alocação de polos que eliminam o valor do *overshoot*.

Outra sugestão para trabalhos futuros é a abordagem de controle multivariável, pois durante o desenvolvimento deste trabalho, observou-se a oportunidade de implementação de um controle de nível e temperatura simultâneos.

Aplicações utilizando controle digital com programação FPGA vêm se destacando devido à possibilidade de processamento paralelo de diversas funções, o que torna isso a principal diferença e vantagem do FPGA quando comparado à diferentes Microcontroladores e DSPs que possuem processamento sequencial. Dessa forma, sugere-se realizar a substituição do microcontrolador Arduino e *software* MATLAB pela utilização do *Field-Programmable Gate Array* (FPGA), e a implementação de ambos os controles utilizados neste trabalho.

REFERÊNCIAS

- AGUIRRE, L. A. **Introdução à Identificação de Sistemas**. 3. ed. Belo Horizonte: UFMG, 2007.
- ALMEIDA, J. L. A. d. **Dispositivos semicondutores: Tiristores**. 11. ed. São Paulo: Editora Érica, 2007.
- ALUTAL, C. I. **Termorresistência PT100**. 2014. Disponível em: <<http://www.pt100sensores.com.br/>>. Acesso em: 16 de out. de 2014.
- ARDUINO, e. C. **Controle de motor CC com o L293D - Ponte H**. 2014. Disponível em: <<http://www.arduinoocia.com.br/2014/04/control-de-motor-cc-com-o-l293d-ponte-h.html>>. Acesso em: 14 de jul. de 2014.
- ASTROM, K.; HAGGLUND, T. **PID Controllers: Theory design and tuning**. 2. ed. [S.l.]: Instrument Society of America, 1995.
- BEZERRA, M. S. **Projeto, Implementação e Ensaio de um Controlador PID utilizando FPGA**. 68 f. Monografia (Graduação em Engenharia Elétrica) — Universidade Federal do Ceará, Fortaleza, 2010.
- CARMO, M. J. d. **Ambiente educacional multifuncional integrado para sintonia e avaliação do desempenho de malhas industriais de controle**. Dissertação (Mestrado) — Centro Federal de Educação Tecnológica Federal do Minas Gerais - CEFET, Juiz de Fora, MG, 2006.
- COELHO, A. A. R.; COELHO, L. d. S. **Identificação de Sistemas Dinâmicos Lineares**. Florianópolis: Editora da UFSC, 2004.
- CYTRON TECHNOLOGIES. **Product User Manual - HC-SR04 Ultrasonic Sensor**. Johor, Malaysia, May 2013. 10 p.
- DATAPOOL, ELETRÔNICA. **Módulo de Nível e Temperatura 2325**: Teoria sistema de controle. Itajubá, 2003. 49 p.
- DATAPOOL, ELETRÔNICA. **Módulo de Nível e Temperatura 2325**: Manual de manutenção e calibração. Itajubá, 2003. 37 p.
- DORF, R. C.; BISHOP, R. H. **Sistemas de Controle Moderno**. 8. ed. Rio de Janeiro: Editora LTC, 2001.
- ECIL. **Sensores PT100**. 2014. Disponível em: <<http://www.ecil.com.br/temperatura-industrial/pirometria/sensores-pt100/>>. Acesso em: 16 de out. de 2014.
- EDTEC. **Medição de temperatura por termoresistência PT100**. 2014. Disponível em: <<http://www.edtec.com.br/termoresistencias%20-%20.htm>>. Acesso em: 16 de out. de 2014.
- FAATESP. **Automação Industrial Temperatura**. 2014. Disponível em: <http://www.faatesp.edu.br/publicacoes/Instrumenta%E7%E3o%20Industrial_Temperatura_pg1_40.pdf>. Acesso em: 28 de nov. de 2014.

MALVINO, A. P. **Eletrônica**. 4. ed. São Paulo: Person Makron Books, 2008.

MARTINS, S. B.; RANGEL, R. F.; MORAIS, . F. d.; SILVA, S. A. d.; LIRA, V. V. Construção de um protótipo para o controle de temperatura. **V Congresso Norte-Nordeste de Pesquisa e Inovação CONNEPI**, Paraíba, 2010.

MCROBERTS, M. **Arduino Básico**. São Paulo: Novatec Editora, 2011.

MENEZES, P. V. F. S.; SILVA, A. N.; PEREIRA I. G. AND MONTEIRO, N. A. B.; SILVA, R. N.; LACERDA, S. L. M. Análise de sensor optoeletrônico no monitoramento de níveis de líquidos em reservatórios. **V Congresso Norte-Nordeste de Pesquisa e Inovação CONNEPI**, Maceio, Alagoas, 2010.

MORAES, C. C.; CASTRUCCI, P. L. **Engenharia de Automação Industrial**. Rio de Janeiro: Editora LTC, 2001.

NISE, N. S. **Engenharia de Sistemas de Controle**. 3. ed. Rio de Janeiro: Editora LTC, 2012.

OGATA, K. **Engenharia de Controle Moderno**. 5. ed. São Paulo: Person Prentice Hall, 2010.

PELZ, G. M.; JUNIOR, P. B.; VARGAS, A. d. N. Implementação de um sistema de controle de nível de dois tanques acoplados. **Congresso Brasileiro de Automática**, Belo Horizonte, p. 7, 2014.

QUEIROGA, S. L. M. d.; MONTEIRO, P. F.; NETO, B. B. O. Aplicabilidade dos microcontroladores em inovações tecnológicas. **Congresso Norte Nordeste de Pesquisa e Inovação**, 2012.

SCHMIDT, A. M. **Controle de Nível de Líquido utilizando controlador Lógico Programável**. 35 f. Monografia (Graduação em Controle e Automação) — Universidade Federal de Ouro Preto, Ouro Preto, 2008.

SMITH, C. A.; CORRIPIO, A. B. **Principles and Practices of Automatic Process Control**. [S.l.]: Wiley, 1997.

ST MICROELECTRONICS. **Dual Full - Bridge Driver L298**. Italy, 2000. 13 p.

UNO. **Uno, Arduino**. 2014. Disponível em: <<http://arduino.cc/en/Main/arduinoBoardUno>>. Acesso em: 15 de out. de 2014.

APÊNDICE A - CÓDIGO ARDUINO 1

Código Implementado no Arduino 1 - Aquisição dos dados do sensor *HC_SR04*

```

# include <LiquidCrystal.h> // Incluir biblioteca do LCD
# define trigPin 10
# define echoPin 7
# define Backlight 6
LiquidCrystal lcd(11, 8, 5, 4, 3, 2);

void setup ()
Serial.begin(9600); //Inicializa a serial
pinMode (9, OUTPUT); //Saída MATLAB
pinMode(trigPin, OUTPUT); // Trig Sensor Nível
pinMode(echoPin, INPUT); //Entrada sensor ECHO

lcd.begin(16,2); //Inicializa LCD
pinMode(Backlight, OUTPUT);
digitalWrite(Backlight, HIGH);
lcd.clear(); //Limpa o LCD
}
void loop ()
float Temp, Dist, nivell, sinl; // definição das variáveis
digitalWrite (trigPin, LOW); // pino TRIG em nível baixo
delayMicroseconds(2);
digitalWrite(trigPin, HIGH); // pino TRIG em nível alto
delayMicroseconds(10);
digitalWrite(trigPin, LOW); // pino TRIG em nível baixo

Temp = pulseIn(echoPin, HIGH); // aquisição dos dados do pino ECHO
Dist = (Temp/2)/ 29.1; // Cálculo da distancia em centímetros
nivell = (20 - Dist); // conversão do sentido dos dados da distância
sinl = (12.74 * nivell); // 255/20 = 12.74
analogWrite (9, sinl); // sinal do pino ECHO enviado para o circuito
RC

Serial.print(Dist); // Mostra na porta serial
Serial.println();
lcd.setCursor(0,0);
lcd.print("Cent.: ");
lcd.print("cm ");
lcd.setCursor(7,0);

```

```
lcd.print(nivel1); } }
```

APÊNDICE B – CÓDIGO ARDUINO 2 - MATLAB

Código Implementado no MATLAB – Estratégias de Controle

B.1 CÓDIGO DO CONTROLE CLÁSSICO PARA O SISTEMA DE NÍVEL

```

close all; clear all; clc
a = arduino('COM6'); // porta que o Arduino 6 esta conectada

Nit=30000;
Tensao = zeros(1, Nit); // Definição das variáveis TensTemp = zeros(1,
Nit);
Nivel=zeros(1,Nit);
v_tempoN=zeros(1, Nit);
saida=zeros(1,Nit);
parc1=zeros(1,Nit);
parc2=zeros(1,Nit);
parc3=zeros(1,Nit);
resto=zeros(1,Nit);
resultado=zeros(1,Nit);
erro_k=zeros(1,Nit);
erro_k1=0;
erro_k2=0;

a.pinMode(3,'OUTPUT'); //Saída PWM enviado para a placa de acionamento
a.pinMode(5,'OUTPUT');

disp('Realizando Amostragem...');
tic;

kp=0.036;
ki=320;
kd=0;
T_samp=0.0330;

// Plota Gráfico em tempo real
figure();clf; h=plot(nan,nan,'b-');grid,xlim([1 Nit]),ylim([0 5]),
grid
xlabel('Amostras')
ylabel('Nível do reservatório')

```

```
grid
```

```

for i = 2:Nit+1 // quantidade de amostras realizadas no ensaio
T(i) = a.analogRead (0); // Leitura do pino A0 do Arduino 2, referente
ao nível
Tensao(i) = (T(i) * 4.887e-3); // Conversão realizada em tensão proporc
ao nível
Nivel=(4 * Tensao); // gráfico em função de centímetros

if i<=10000
s_point= 3.5;
end
if i<=20000 && i>10000
s_point= 2.5;
end
if i<=Nit+1 && i>20000
s_point= 1.5;
end
valorEntrada(i)=Tensao(i);
erro_k(i)= s_point - valorEntrada(i);
parc1(i)=kp*(erro_k(i));
parc2(i)=((ki*erro_k(i)) + parc2(i-1));

if parc2(i) >2
parc2(i) = 2;
end
if parc2(i) < -2
parc2(i) = -2;
end
saida(i)= parc1(i) + parc2(i);
resto(i)= mod(saida(i)255,5);

resultado(i) =((saida(i)255 - resto(i))/5);

if resultado(i) >= 255
resultado(i) = 255;
else
if resultado(i) <= 0
resultado(i)=0;
end

a.analogWrite (3, resultado(i));
a.digitalWrite (5, 1);
end

```

```

pause(1e6);
set(h,'xdata',[1:i],'ydata',[Tensao([1:i])]);
end
end

a.digitalWrite (5, 0);
a.analogWrite (3, 0);

tempo=toc
v_tempoN=linspace(0,tempo,Nit); // cria um vetor de tempo considerando
um periodo amostral igual a medida entre o tempo total e o numero
de amostras.
samprate = Nit/toc
t= 1/samprate

disp('Fim da amostragem.');
```

disp('Pressione qualquer tecla para encerrar o programa.');

B.2 CÓDIGO DO CONTROLE CLÁSSICO PARA O SISTEMA DE TEMPERATURA

```

close all; clear all ; clc
a = arduino('COM6'); // porta que o Arduino 6 esta conectada

Nit2=90000;
Tempfinal=zeros(1, Nit2);
v_tempoT=zeros(1, Nit2);
saida=zeros(1,Nit2);
parc1=zeros(1,Nit2);
parc2=zeros(1,Nit2);
parc3=zeros(1,Nit2);
resto=zeros(1,Nit2);
resultado=zeros(1,Nit2);
erro_k=zeros(1,Nit2);
erro_k1=0;
erro_k2=0;
Diferenca = zeros(1, Nit2);
Diferenca2 = zeros(1, Nit2);
Temperatura = zeros(1, Nit2);
Temperatura2 = zeros(1, Nit2);

a.pinMode (3,'OUTPUT'); //Saída PWM enviado para a placa de acionamento
```

```

a.pinMode(5,'OUTPUT'); // Pino utilizado para a bomba não ser acionada
no momento errado
a.pinMode (9,'OUTPUT'); //Saída PWM enviado para a placa de acionamento

disp('Realizando Amostragem...');
tic;
kpT=0.01586;
kiT=10;
kdT=0;

figure();clf; hh=plot(nan,nan,'b-');grid,xlim([1 Nit2]),ylim([10
50]),
grid
xlabel('Amostras')
ylabel('Temperatura em Graus Celsius')

for i = 2:Nit2+1 // quantidade de amostras realizadas no ensaio
ValorA(i) = a.analogRead (1); // Leitura do pino A0 do Arduino 2,
referente ao nível
ValorB(i) = a.analogRead (2);
Pot(i)= a.analogRead (5);
Diferenca2(i) = (ValorB(i) - ValorA(i));
Diferenca(i)= Diferenca2(i) * 4.887e-3; // Passar para tensão
Temperatura(i)= ((Diferenca(i)*(400/5))); // resistencia
Temperatura(i)= (Temperatura(i)*100)/38.5;

if i<=40000
s_point= 0.14;
end
if i>40000
s_point= 0.19;
end

valorEntrada(i)=Diferenca(i);
erro_k(i)= 15*(s_point - valorEntrada(i)); //calculo do erro em valores
de tensão

parc1(i)=kpT*(erro_k(i));
parc2(i)=((kiT*erro_k(i)) + parc2(i-1));

if parc2(i) >2
parc2(i) = 2;
end
if parc2(i) < -2

```

```

parc2(i) = -2;
end
saida(i)= parc1(i) + parc2(i);
resto(i)= mod(saida(i)*255,5);
resultado(i) =((saida(i)*255 - resto(i))/5);

if resultado(i) >= 10
resultado(i) = 10;
else
if resultado(i) <= 0
resultado(i)=0;
end
end

a.analogWrite (10, resultado(i));
end

pause(1e-6);
set(hh,'xdata',[1:i],'ydata',[Temperatura([1:i])]);
end
end

a.digitalWrite (5, 0);
a.analogWrite (3, 0);

tempo=toc
v_tempoT=linspace(0,tempo,Nit2);
samprate = Nit/toc
t= 1/samprate
disp('Fim da amostragem.');
```

disp('Pressione qualquer tecla para encerrar o programa.');

B.3 CÓDIGO DO CONTROLE MODERNO PARA O SISTEMA DE NÍVEL

```

close all; clear all; clc
a = arduino('COM6'); // porta que o Arduino 6 esta conectada

Nit=80000;
Tensao = zeros(1, Nit);
TensTemp = zeros(1, Nit); Nivel=zeros(1,Nit);
v_tempoN=zeros(1, Nit);
saida=zeros(1,Nit);
parc1=zeros(1,Nit);
parc2=zeros(1,Nit);
```



```

parc3=zeros(1,Nit);
resto=zeros(1,Nit);
resultado=zeros(1,Nit);
erro_k=zeros(1,Nit);
erro_k1=0;
erro_k2=0;
vecV=zeros(1,Nit);
vecP=zeros(1,Nit);
soma_a2=zeros(1,Nit);
soma_a1=zeros(1,Nit);
x=zeros(3,Nit);
u=zeros(1,Nit);
Vel=zeros(1,Nit);
Bf(1)=5.9796e-005; Bf(2)= 2.9898e-004; Bf(3)= 5.9796e-004; Bf(4)=
5.9796e-004; Bf(5)= 2.9898e-004; Bf(6)= 5.9796e-005;
Af(1)=1; Af(2)= -3.9845; Af(3)= 6.4349; Af(4)= -5.2536; Af(5)= 2.1651;
Af(6)= -0.3599;

a.pinMode(3,'OUTPUT'); //Saída PWM enviado para a placa de acionamento
a.pinMode(5,'OUTPUT');

disp('Realizando Amostragem...');
tic;
// Gera gráfico em tempo real do nível
figure();clf; h=plot(nan,nan,'b-');grid,xlim([1 Nit]),ylim([0 20]),
grid
xlabel('Amostras')
ylabel('Nível')
grid

// Gera gráfico em tempo real da velocidade
figure();clf; h3=plot(nan,nan,'b-');grid,xlim([0 Nit]),ylim([-0.5
0.5]),
grid
xlabel('Amostras')
ylabel('Velocidade')
grid
j = 1;
N =1500;

// PARAMETROS DE CONTROLE MODERNO - GANHO
// K = [924.9 9631.7 -40.7]; J=[-0.8 -0.1 -0.5] SINAL
// K = [2478 23885 -163]; J=[-0.8 -0.4 -0.5] SINAL
// K = [4031 38138 -285]; J=[-0.8 -0.5 -0.7] SINAL

```

```

Tempo = 0.0521;
eta = 0;
Pos = zeros(1,6);

for i = 1:Nit // quantidade de amostras realizadas no ensaio
T(i) = a.analogRead (0); // Leitura do pino A0 do Arduino 2, referente
ao nível
Tensao(i) = (T(i) * 4.887e-3); // Conversão realizada em tensão proporca
ao nível
Nivel(i)=(4 * Tensao(i)); // gráfico em função de centímetros

if (i >= 6)
Pos(i) = -Af(2)*Pos(i-1) - Af(3)*Pos(i-2) - Af(4)*Pos(i-3) - Af(5)*Pos(i-4)
- Af(6)*Pos(i-5) + Bf(1)*Tensao(i) + Bf(2)*Tensao(i-1) + Bf(3)*Tensao(i-2)
+ Bf(4)*Tensao(i-3) + Bf(5)*Tensao(i-4) + Bf(6)*Tensao(i-5);
Vel(i) = (Pos(i)-Pos(i-1))/Tempo;

if i<=25000
r= 2.5;
end
if i<=50000 && i>25000
r= 3.5;
end
if i<=Nit && i>50000
r= 1.5;
end

eta = eta + (r-Tensao(i))*Tempo;
etameu(i) = eta;
x(:,i)=[Tensao(i); Vel(i); eta];
u(i) = -K*x(:,i);

resto(i)= mod(u(i)*255,5);
resultado(i) =((u(i)*255 - resto(i))/5);

if resultado(i) >= 255
resultado(i) = 255;
else
if resultado(i) <= 0
resultado(i)=0;
end
end
end

```

```
a.analogWrite (3, resultado(i));
a.digitalWrite (5, 1);

pause(1e-6);
set (h3,'xdata',[1:i],'ydata',[Vel([1:i]))];
end

pause(1e-6);
set (h,'xdata',[1:i],'ydata',[Nivel([1:i]))];
end
end

a.digitalWrite (5, 0);
a.analogWrite (3, 0);

tempo=toc
v_tempoN=linspace(0,tempo,Nit);
samprate = Nit/toc
t= 1/samprate

disp('Fim da amostragem.');
```

```
disp('Pressione qualquer tecla para encerrar o programa.');
```

APÊNDICE C - CÓDIGO ARDUINO 3

Código Implementado no Arduino 3 - Sincronismo com a rede

```
#include <avr/io.h>
#include <avr/interrupt.h>

#define DETECT 2 //zero cross detect
#define GATE 9 //triac gate
#define PULSE 4 //trigger pulse width (counts)
int sensor = 5; // int valorsensor = 0; // int potencia = 0; // //int
frequencia=120000;

void setup()
Serial.begin(9600);
pinMode(DETECT, INPUT); //zero cross detect
digitalWrite(DETECT, HIGH); //enable pull-up resistor
pinMode(GATE, OUTPUT); //triac gate control

OCR1A = 100; //initialize the comparator
TIMSK1 = 0x03; //enable comparator A and overflow interrupts
TCCR1A = 0x00; //timer control registers set for
TCCR1B = 0x00; //normal operation, timer disabled

// set up zero crossing interrupt
attachInterrupt(0, zeroCrossingInterrupt, CHANGE);

void zeroCrossingInterrupt() //zero cross detect
TCCR1B=0x04; //start timer with divide by 256 input
TCNT1 = 0x04; //reset timer - count from zero

ISR(TIMER1_COMPA_vect) //comparator match
digitalWrite(GATE,HIGH); //set triac gate to high
TCNT1 = 120000-PULSE; //trigger pulse width

ISR(TIMER1_OVF_vect) //timer1 overflow
digitalWrite(GATE,LOW); //turn off triac gate
TCCR1B = 0x00; //disable timer stopd unintended triggers
```

```
void loop() // sample code to exercise the circuit

valorsensor = analogRead(sensor); // valores analógicos de 0 a 1023

Serial.println(valorsensor);
potencia = valorsensor/4;

analogWrite(9, potencia);
```

ANEXO A - CÓDIGO ARDUINO 2

Código Implementado no Arduino 2 - Interface para o uso do MATLAB

```

/* Analog and Digital Input and Output Server for MATLAB */
/* Giampiero Campa, Copyright 2013 The MathWorks, Inc */
/* This file is meant to be used with the MATLAB arduino IO
package, however, it can be used from the IDE environment
(or any other serial terminal) by typing commands like:

0e0 : assigns digital pin #4 (e) as input
0f1 : assigns digital pin #5 (f) as output
0n1 : assigns digital pin #13 (n) as output
1c : reads digital pin #2 (c)
1e : reads digital pin #4 (e)
2n0 : sets digital pin #13 (n) low
2n1 : sets digital pin #13 (n) high
2f1 : sets digital pin #5 (f) high
2f0 : sets digital pin #5 (f) low
4j2 : sets digital pin #9 (j) to 50=ascii(2) over 255
4jz : sets digital pin #9 (j) to 122=ascii(z) over 255
3a : reads analog pin #0 (a)
3f : reads analog pin #5 (f)

R0 : sets analog reference to DEFAULT
R1 : sets analog reference to INTERNAL
R2 : sets analog reference to EXTERNAL
X3 : roundtrip example case returning the input (ascii(3))
99 : returns script type (0 adio.pde ... 3 motor.pde ) *

/* define internal for the MEGA as 1.1V (as as for the 328) */
#if defined(__AVR_ATmega1280__) defined(__AVR_ATmega2560__)
#define INTERNAL INTERNAL1V1
#endif

void setup()
/* initialize serial
Serial.begin(115200);

void loop()

```

```

/* variables declaration and initialization */
static int s = -1; /* state */
static int pin = 13; /* generic pin number */
int val = 0; /* generic value read from serial */
int agv = 0; /* generic analog value */
int dgv = 0; /* generic digital value */
/* The following instruction constantly checks if anything
is available on the serial port. Nothing gets executed in
the loop if nothing is available to be read, but as soon
as anything becomes available, then the part coded after
the if statement (that is the real stuff) gets executed */

if (Serial.available() >0)
/* whatever is available from the serial is read here */
val = Serial.read();

/* This part basically implements a state machine that
reads the serial port and makes just one transition
to a new state, depending on both the previous state
and the command that is read from the serial port.
Some commands need additional inputs from the serial
port, so they need 2 or 3 state transitions (each one
happening as soon as anything new is available from
the serial port) to be fully executed. After a command
is fully executed the state returns to its initial
value s=-1 */
switch (s)
/* s=-1 means NOTHING RECEIVED YET ***** */
case -1:
if (val>47 && val<90)
/* the first received value indicates the mode
49 is ascii for 1, ... 90 is ascii for Z
s=0 is change-pin mode;
s=10 is DI; s=20 is DO; s=30 is AI; s=40 is AO;
s=90 is query script type (1 basic, 2 motor);
s=340 is change analog reference;
s=400 example echo returning the input argument;
s=10*(val-48);

/* the following statements are needed to handle
unexpected first values coming from the serial (if
the value is unrecognized then it defaults to s=-1) */
if ((s>40 && s<90) || (s>90 && s=340 && s=400))

```

```

s=-1;

/* the break statements gets out of the switch-case, so
/* we go back and wait for new serial data */
break; /* s=-1 (initial state) taken care of */
/* s=0 or 1 means CHANGE PIN MODE */

case 0:
/* the second received value indicates the pin
from abs('c')=99, pin 2, to abs('')=166, pin 69 */
if (val>98 && val<167)
pin=val-97; /* calculate pin */
s=1; /* next we will need to get 0 or 1 from serial */

else
s=-1; /* if value is not a pin then return to -1 */

break; /* s=0 taken care of */

case 1: /* the third received value indicates the value 0 or 1 */
if (val>47 && val<50)
/* set pin mode */
if (val==48)
pinMode(pin,INPUT);

else
pinMode(pin,OUTPUT);

s=-1; /* we are done with CHANGE PIN so go to -1 */
break; /* s=1 taken care of */
/* s=10 means DIGITAL INPUT ***** */

case 10:
/* the second received value indicates the pin
from abs('c')=99, pin 2, to abs('')=166, pin 69 */
if (val>98 && val<167)
pin=val-97; /* calculate pin */
dgv=digitalRead(pin); /* perform Digital Input */
Serial.println(dgv); /* send value via serial */

s=-1; /* we are done with DI so next state is -1 */
break; /* s=10 taken care of */
/* s=20 or 21 means DIGITAL OUTPUT ***** */

```



```

case 20:
/* the second received value indicates the pin
from abs('c')=99, pin 2, to abs('')=166, pin 69 */
if (val>98 && val<167)
pin=val-97; /* calculate pin */
s=21; /* next we will need to get 0 or 1 from serial */

else
s=-1; /* if value is not a pin then return to -1 */

break; /* s=20 taken care of */

case 21:
/* the third received value indicates the value 0 or 1 */
if (val>47 && val<50)
dgv=val-48; /* calculate value */
digitalWrite(pin,dgv); /* perform Digital Output */

s=-1; /* we are done with DO so next state is -1 */
break; /* s=21 taken care of */
/* s=30 means ANALOG INPUT ***** */

case 30:
/* the second received value indicates the pin
from abs('a')=97, pin 0, to abs('p')=112, pin 15 */
if (val>96 && val<113)
pin=val-97; /* calculate pin */
agv=analogRead(pin); /* perform Analog Input */
Serial.println(agv); /* send value via serial */

s=-1; /* we are done with AI so next state is -1 */
break; /* s=30 taken care of */
/* s=40 or 41 means ANALOG OUTPUT ***** */

case 40:
/* the second received value indicates the pin
from abs('c')=99, pin 2, to abs('')=166, pin 69 */
if (val>98 && val<167)
pin=val-97; /* calculate pin */
s=41; /* next we will need to get value from serial */

else
s=-1; /* if value is not a pin then return to -1 */

```

```

break; /* s=40 taken care of */

case 41:
/* the third received value indicates the analog value */
analogWrite(pin,val); /* perform Analog Output */
s=-1; /* we are done with AO so next state is -1 */
break; /* s=41 taken care of */
/* s=90 means Query Script Type:
(0 adio, 1 adioenc, 2 adiosrv, 3 motor) */

case 90:
if (val==57)
/* if string sent is 99 send script type via serial */
Serial.println(0);

s=-1; /* we are done with this so next state is -1 */
break; /* s=90 taken care of */
/* s=340 or 341 means ANALOG REFERENCE ***** */

case 340:
/* the second received value indicates the reference,
which is encoded as is 0,1,2 for DEFAULT, INTERNAL
and EXTERNAL, respectively. Note that this function
is ignored for boards not featuring AVR or PIC32 */

#ifdef __AVR__
#ifdef __PIC32MX__
switch (val)

case 48:
analogReference(DEFAULT);
break;

case 49:
analogReference(INTERNAL);
break;

case 50:
analogReference(EXTERNAL);
break;
default: /* unrecognized, no action */
break;

```

```
        # endif
s=-1; /* we are done with this so next state is -1 */
break; /* s=341 taken care of */
/* s=400 roundtrip example function (returns the input)*/

case 400:
/* the second value (val) can really be anything here */
/* This is an auxiliary function that returns the ASCII
value of its first argument. It is provided as an
example for people that want to add their own code */
/* your own code goes here instead of the serial print */
Serial.println(val);
s=-1; /* we are done with the aux function so -1 */
break; /* s=400 taken care of */

/* ***** UNRECOGNIZED STATE, go back to s=-1 ***** */
default:
/* we should never get here but if we do it means we
are in an unexpected state so whatever is the second
received value we get out of here and back to s=-1 */
s=-1; /* go back to the initial state, break unneeded */
/* end switch on state s */
/* end if serial available */
/* end loop statement */
```