

TRABALHO DE GRADUAÇÃO

CONTROLE TITO VIA DESACOPLAMENTO DE MALHAS
DE CLIMATIZAÇÃO HÍBRIDA

Artur Macedo Araújo

Brasília, Dezembro de 2016



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO
**CONTROLE TITO VIA DESACOPLAMENTO DE MALHAS
DE CLIMATIZAÇÃO HÍBRIDA**

Artur Macedo Araújo

*Relatório submetido como requisito parcial de obtenção
de grau de Engenheiro de Controle e Automação*

Banca Examinadora

Prof. Dr. Adolfo Bauchspiess, ENE/UnB

Orientador

Prof. Dr. Marco A. F. do Egito Coelho,

ENE/UnB

Examinador interno

Prof. Dr Renato A. Borges, ENE/UnB

Examinador interno

Brasília, Dezembro de 2016

FICHA CATALOGRÁFICA

ARTUR MACEDO ARAÚJO

Controle TITO via desacoplamento de malhas de sistema de climatização híbrido

[Distrito Federal] 2016.

x, 101p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2016). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.

1. Controle TITO

2. Sistemas híbridos

3. Eficiência energética

4. Desacoplamento

I. Mecatrônica/FT/UnB

II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

ARAÚJO, A. M. (2016). Controle TITO via desacoplamento de malhas de sistema de climatização híbrido. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-n°022, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 101p.

CESSÃO DE DIREITOS

AUTOR: Artur Macedo Araújo

Controle TITO via desacoplamento de malhas de sistema de climatização híbrido: implementação de controle de temperatura e umidade em um sistema de climatização híbrida

GRAU: Engenheiro de Controle e Automação

ANO: 2016

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Artur Macedo Araújo

SQS 415 Bloco K - Asa Sul

70298-110 Brasília – DF – Brasil.

Dedicatória

Dedico este trabalho aos meu pais, Deusélia e Ribamar, pelo incondicional apoio e infín-dável paciência. À minha irmã, Marina, por nunca ter deixado de acreditar

Artur Macedo Araújo

Agradecimentos

Quero agradecer a todos que de maneira direta ou indireta me ajudaram durante esses anos de graduação. Em especial ao Prof. Adolfo Bauchspiess, meu orientador, por seus valiosos ensinamentos, paciência e disposição que foram essenciais para a conclusão desse trabalho. A todos os mestres da UnB, que de alguma forma contribuíram para meu crescimento tanto acadêmico quanto pessoal. Aos meus colegas de curso, sem os quais esta trajetória teria sido impossível. Aos meus grandes amigos, Matheus e Marcos, por patilharem as derrotas e vitórias da engenharia e pelos bons momentos, mesmo nas piores horas.

Agradecimento especial à Na, por todo seu amor mostrado em forma de carinho, apoio e paciência. Por fim, porém mais importante, agradeço à minha família, fonte de amor incondicional, que são a pedra angular que dá sentido à minha existência.

Artur Macedo Araújo

RESUMO

Existe atualmente uma crescente preocupação, e exigência, relacionada ao conforto térmico associado à qualidade de ar em um determinado ambiente.

Neste contexto, sistemas convencionais de climatização empregados em larga escala tanto em ambientes residenciais quanto prediais e industriais, embora tenham boa eficiência térmica, consomem uma quantidade de energia elétrica considerável, especialmente levando em consideração os parâmetros de consumo desejados para a atualidade.

Este trabalho visa projetar um controle para sistemas de climatização híbrida, em especial o que se encontra no Laboratório de Robótica e Automação, que utiliza um conjunto de modos condensador e evaporativo acoplados, desta forma deseja-se implementar uma técnica de controle que dê suporte à estes dois modos, resultando no menor consumo energético possível.

Ao que concerne o controle, deve-se controlar um sistema TITO via desacoplamento. Essa implementação se dará utilizando módulos Arduino e Raspberry Pi; com transmissão de dados tanto cabeada e via rede. Modelagem e processamento foram feitos com auxílio do Matlab.

Palavras-chave: Controle, multi-malhas, híbrido, desacoplamento, TITO.

ABSTRACT

Lately, there is a growing concern and demand related to thermal comfort associated with air quality in a particular environment.

In this context, conventional air conditioning systems are used in large scale both in residential as in building and industrial environments. Although it has good thermal efficiency, consumes a considerable amount of electricity, especially considering the current desired consumption parameters.

This study aims designing a control for hybrid air conditioning systems, in particular the one lying on Robotics and Automation Lab, which uses a set of condenser and evaporative systems coupled; therefore is desired to implement some control technique that may support these two sets, resulting in the lowest possible energy consumption.

Concerning the control, a decoupling strategy will be used to control a TITO system. This implementation will be made using Arduino modules and raspberry pi; with data transmission both wired and through network. Modeling and processing were done with the support of Matlab.

Palavras-chave: Control, multi-mesh, hybrid, decoupling, TITO

SUMÁRIO

1	INTRODUÇÃO	1
1.1	MOTIVAÇÃO	1
1.2	OBJETIVOS	2
1.3	TRABALHOS ANTERIORES	2
2	FUNDAMENTAÇÃO TEÓRICA	3
2.1	CONFORTO TÉRMICO	3
2.1.1	PMV- ÍNDICE DO CONFORTO TÉRMICO	4
2.2	PROPRIEDADES PSICOMÉTRICAS	6
2.2.1	PRESSÕES PARCIAIS	6
2.2.2	UMIDADE ABSOLUTA	7
2.2.3	UMIDADE RELATIVA	8
2.2.4	TEMPERATURA DE BULBO SECO E TEMPERATURA DE BULBO ÚMIDO	8
2.2.5	TEMPERATURA DE ORVALHO	9
2.3	CARTA PSICOMÉTRICA	9
3	AUTOMAÇÃO DO SISTEMA DE CLIMATIZAÇÃO HÍBRIDA	12
3.1	SISTEMA DE CLIMATIZAÇÃO HÍBRIDO DO LARA	12
3.1.1	FUNCIONAMENTO CONVENCIONAL	12
3.1.2	FUNCIONAMENTO EVAPORATIVO	13
3.2	OPENHAB	15
3.3	MQTT	17
3.4	MATERIAIS	18
3.4.1	SENSOR DHT22	18
3.4.2	RASPBERRY PI	20
3.4.3	ARDUINO	21
3.4.4	MÓDULO ETHERNET ENC28J60	22
3.4.5	MÓDULO RELÉ	23
3.5	PWM	24
4	CONTROLE TITO	26
4.1	MODELO DO SISTEMA	26
4.1.1	SIMULAÇÃO MODELO	29

4.2	SISTEMA TITO	30
4.3	MATRIZ DE GANHOS RELATIVOS	31
4.4	DESACOPLAMENTO	33
4.4.1	DESACOPLADOR IDEAL.....	34
4.4.2	DESACOPLADOR SIMPLIFICADO.....	35
4.4.3	DISCRETIZAÇÃO DESACOPLADOR.....	37
4.5	PROJETO CONTROLADOR.....	37
4.5.1	CONTROLADOR PI.....	38
4.5.2	CONTROLE UMIDADE.....	40
5	RESULTADOS	42
5.1	TEMPERATURA	42
5.2	UMIDADE	49
6	CONCLUSÕES	53
6.1	PERSPECTIVAS FUTURAS	54
	REFERÊNCIAS BIBLIOGRÁFICAS	55
	ANEXOS.....	57
I	PROGRAMAS UTILIZADOS	60
I.1	PROGRAMAS MATLAB.....	60
I.2	PROGRAMA ARDUINO	62

LISTA DE FIGURAS

2.1	Representação esquemática da fisiologia humana e trocas térmicas	4
2.2	PPD <i>versus</i> PMV [1]	5
2.3	Lei de Dalton	7
2.4	Psicômetro [2]	9
2.5	Carta psicométrica (flycarpet.net/en/psyonline)	10
2.6	Propriedades da carta psicométrica	10
3.1	Ciclo de refrigeração [3]	13
3.2	Sistema evaporativo [4].....	14
3.3	Esquemático openHAB.....	16
3.4	GUI openHAB.....	16
3.5	Protocolo MQTT	18
3.6	Sensor DHT22 (adafruit.com)	19
3.7	Raspberry Pi (raspberrypi.org)	20
3.8	Arduino Mega (arduino.cc)	21
3.9	Módulo Ethernet ENC28J60 (blog.filipeflop.com/)	22
3.10	Módulo com 4 relés (Modelo: SRD-05VDC0SL-C)	23
3.11	Exemplo de PWM aplicado ao Arduino	24
3.12	Diagrama completo de conexão	25
4.1	Sala de reuniões do LARA [5].....	26
4.2	Modelo convencional	27
4.3	Modelo evaporativo	28
4.4	Modelo completo.....	28
4.5	Resultado da simulação para temperatura	29
4.6	Resultado da simulação para umidade	30
4.7	Estrutura TITO.....	31
4.8	Sistema multivariável com desacoplador completo	34
4.9	Sistema multivariável com desacoplador simples	35
4.10	Diagrama de blocos no Simulink com desacoplador	36
4.11	LGR da malha de temperatura	39
4.12	Unidade de controle Simulink	39
4.13	Simulação controle PI para temperatura	40

5.1	Comportamento do controle de temperatura	42
5.2	Comparação temperatura interna/externa	43
5.3	Comparação temperatura interna/TBU externa.....	44
5.4	Efeito do modo evaporativo na temperatura da sala de reuniões	45
5.5	Variação temperatura da sala de reuniões e umidade externa	46
5.6	Interface de controle	46
5.7	Estratégia 2 para controle de temperatura da sala de reuniões.....	47
5.8	Média móvel da temperatura interna da sala de reuniões.....	48
5.9	Comportamento da umidade interna da sala de reuniões	50
5.10	Comparação da umidade interna da sala de reuniões, umidade externa e do Lara	51
5.11	Comparação da umidade interna da sala de reuniões, umidade externa e do Lara, experimento 2.....	52
A.I	Esquemático da conexão	59

LISTA DE TABELAS

2.1	Trocas térmicas	4
2.2	Padrão ASHRAE	5
2.3	Parâmetros de conforto umidade [6]	8
3.1	Métodos do MQTT	17
3.2	Pinagem do DHT22	19
3.3	Especificações do Arduino Mega 2560	21
3.4	Conexão Módulo	22
4.1	Valores de λ_{ij} [7]	33
5.1	Redução de temperatura por resfriamento evaporativo	44
5.2	Utilização do compressor e bomba d'água	48
5.3	Consumo dos equipamento [8]	49
1	Sensor DHT22 (ccspicc.blospot.com)	58

LISTA DE SÍMBOLOS

v_a	Velocidade do ar	[m/s]
T	Temperatura do ar	[°C]
W	Trabalho externo	[W/m ²]
M	Taxa de produção metabólica	[W/m ²]
m_a	Massa de ar seco	[kg _a]
m_v	Massa de vapor de água	[kg _v]
p_a	Pressão do ar seco	[N/m ²]
p_v	Pressão do vapor de água	[N/m ²]
p_t	Pressão total	[N/m ²]

Símbolos Gregos

ω	Velocidade angular	[rad/s]
λ_{ij}	Ganho relativo da entrada i com saída j	
Λ	Matriz de ganhos relativos	

Grupos Adimensionais

i, k	Contador	
w_{abs}	Umidade absoluta	
ϕ_{ar}	Umidade relativa	
$D_{ij}(s)$	Desacoplador da entrada i para saída j	
$D_{ij}^z(z)$	Desacoplador discreto da entrada i para saída j	
$G_{ij}(s)$	Função de transferência da entrada i para saída j	
$Q_{ij}(s)$	Função de transferência aparente da entrada i para saída j	

Subscritos

ref	referência
fer	ferramenta
sis	sistema
des	desejado

Sobrescritos

- Variação temporal
- Valor médio

Siglas

ASHARE	<i>American Society of Heating, Refrigerating and Air Conditioning Engineers</i>
I/O	<i>Input/Output</i> - Entrada/Saída
IoT	<i>Internet of Things</i> - Internet das Coisas
LARA	Laboratório de Robótica e Automação
LGR	Lugar Geométrico das Raízes
M2M	<i>Machine-to-machine</i>
MIMO	<i>Multiple Input Multiple Output</i>
MQTT	<i>Message Queue Telemetry Transport</i>
OpenHAB [®]	<i>Open Home Automation Bus</i>
PI	Controlador proporcional integrativo
PID	Controlador proporcional integrativo derivativo
PMV	<i>Predicted mean vote</i> - Voto médio previsível
PPD	<i>Predicted Percentage of Dissatisfied</i> - Percentagem Previsível de Insatisfeitos
PWM	<i>Pulse Width Modulation</i> - Modulação por Largura de Pulso
RGA	<i>Relative Gain Array</i>
RPi	Raspberry Pi
SISO	<i>Single Input Single Output</i>
TBS	Temperatura de bulbo seco
TBU	Temperatura de bulbo úmido
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i> - Protocolo de Controle de Transmissão/Protocolo Internet
TITO	<i>Two-Inputs-Two-Outputs</i>
CPU	Unidade Central de Processamento - <i>Central Processing Unit</i>
USB	<i>Universal Serial Bus</i>

Capítulo 1

Introdução

Capítulo que aborda contextualização do tema, importância e motivação, assim como os objetivos deste trabalho.

1.1 Motivação

Ambientes bem climatizados é uma realidade, ou desejo, para boa parte das pessoas, quer para aquecer ambientes com o uso de aquecedores ou para resfriá-los com condicionadores de ar por exemplo. Em países tropicais como o Brasil se faz necessário o uso de climatizadores de ambiente durante várias horas devido às altas temperaturas ao longo de boa parte do ano.

O uso desses tipos de equipamentos que visam o conforto térmico cresceram bastante ultimamente com a queda de preço desses aparelhos e o acesso facilitado. Em ambientes residenciais, prediais e industriais o emprego desses climatizadores demanda bastante energia elétrica, demanda a qual nem sempre pode ser suportada pela matriz energética.

Uma expansão da matriz energética nem sempre é viável economicamente, além dos vários impactos que se pode causar, em especial impactos ambientais. Neste contexto que se encontram os ambientes inteligentes, tendo como um dos objetivos principais o uso racional da energia, proporcionando o máximo de economia e reduzindo consideravelmente os danos ambientais, associado ao conforto do ser humano [5].

Um sistema de climatização híbrido, como o que se encontra na sala de reuniões do LARA, é uma boa alternativa para ambientes inteligentes prediais que devem ter suas aplicações estendidas no futuro. Esse sistema, que utiliza a compressão convencional do gás em conjunto com um método evaporativo, já se mostrou eficiente e vantajoso em relação às formas de resfriamento puramente convencionais, como mostrado por [6].

Entretanto, o controle desse sistema tem sido feito até hoje por meio de simples controladores liga-desliga temporizados que alternam entre sistema convencional e evaporativo a cada intervalo de tempo, ou técnicas heurísticas de controle. Controlar temperatura e umidade pode se tornar um trabalho complexo e exaustivo; conseqüentemente a automação desses processos muitas vezes são feitas de forma empírica; de maneira simplificada.

Esse métodos se mostraram valorosos; o próximo passo então seria um controle projetado levando em consideração a dinâmica do processo de resfriamento da sala de reunião juntamente com a umidade e sua influência no processo de resfriamento. Um entendimento dessas duas malhas, de temperatura e umidade, é fundamental para a implementação de técnicas de controle que levem finalmente à um ganho na eficiência energética do processo como um todo.

Portanto procura-se entender o processo de resfriamento de forma global levando-se em conta o controle de duas variáveis (temperatura e umidade) buscando um conforto térmico com um racional uso da energia.

1.2 Objetivos

O objetivo deste trabalho é projetar um controlador e controlar um sistema de climatização híbrida presente na sala de reuniões do LARA com o intuito que ofereça razoável conforto térmico à todos os ocupantes com um satisfatório consumo de energia elétrica; tendo como resultado um processo que seja mais eficiente energeticamente que os resultados obtidos com as técnicas de controle previamente implementadas.

Para a compreensão do sistema, utilizando-se de métodos fundamentais como a base que rege o processo de refrigeração; deve-se fazer a instrumentação conectando-se sensores para mensurar temperatura e umidade, módulos de processamento dos dados e implementação do controle assim como atuadores para o sistema.

1.3 Trabalhos anteriores

Alguns trabalhos já foram feitos relacionados ao sistema de climatização híbrido da sala de reuniões do LARA, como [6] que foi o primeiro trabalho relacionado ao tema; assim como [5] que foi responsável pelo levantamento do modelo da sala e [9] com uma proposta de controle; todos serviram como base para este trabalho. Este projeto então se coloca como uma contribuição para esse tema já abordado anteriormente, servindo de comparação e com intuito de agregar conhecimento.

Capítulo 2

Fundamentação Teórica

Este capítulo aborda a fundamentação teórica de conceitos importantes para a compreensão do sistema de climatização híbrida

2.1 Conforto térmico

Não se pode tentar definir conforto térmico como algo simples e totalmente teórico visto que depende não somente de fatores físicos, mensuráveis e computáveis, como também de fatores pessoais que portanto variam de indivíduo para indivíduo. Esse fator subjetivo é um ponto que deve ser levado em conta na tentativa de conceituação de conforto térmico.

De acordo com Hensen, conforto térmico pode ser definido como um estado no qual não há impulso de condução para corrigir o comportamento do ambiente [10]. Já a ASHARE (The American Society of Heating, Refrigerating and Air-Conditioning Engineers) define tal termo como sendo a condição da mente em que esta expressa satisfação com o ambiente térmico [11]. Portanto, tal percepção de satisfação será influenciada pelas diferenças naturais de humor, cultura e até sociais de cada indivíduo.

Sensações térmicas são diferentes entre as pessoas, por mais que estejam no mesmo ambiente. Embora sensores meçam mesmos resultados independentemente do local, este não é o caso para as pessoas. Na verdade, pessoas localizadas em espaços semelhantes, submetidos ao mesmo clima, e pertencentes a uma cultura comum, podem emitir opiniões bem diferentes em relação ao conforto térmico devido à combinação de um grande número de fatores que afetam a percepção de cada ser humano [12].

Analisando do ponto de vista fisiológico, os seres humanos são animais homeotérmicos e tendem a manter a temperatura interna constante independente do ambiente em que se encontram; dessa forma assume-se como confortável o ambiente cujas condições permitam manter a temperatura interna constante sem esforços, alcançando um equilíbrio térmico com o meio [5].

Existe atualmente duas principais abordagens para conforto térmico: o modelo adaptativo e o modelo racional, baseado no trabalho de Fanger [13] que propôs que seis variáveis principais regem o conforto térmico: atividade, vestuário, temperatura de bulbo seco do ar, temperatura

média radiante, velocidade relativa do ar e umidade relativa do ar; sendo divididas em variáveis individuais e ambientais [6].

2.1.1 PMV- índice do conforto térmico

De acordo com o estudo de Fanger, visto na Seção 2.1, e as variáveis relacionada ao conforto térmico, sabe-se que a troca de calor realizada pelo corpo pode ocorrer de quatro maneiras: evaporação, convecção, condução e radiação; conforme mostrado na Figura 2.1. No caso da refrigeração do corpo, as principais formas para a redução da temperatura são através da respiração e transpiração.

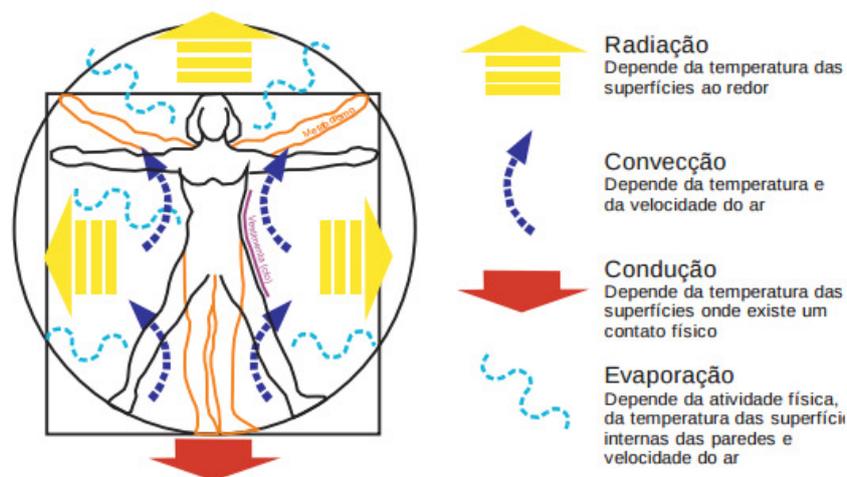


Figura 2.1: Representação esquemática da fisiologia humana e trocas térmicas

O modelo utilizado na Norma Internacional ISO 7730 utiliza o *estado estacionário* desenvolvido por Fanger, o qual assume que o corpo em determinado ambiente encontra-se em estado de equilíbrio, não ocorrendo portanto acúmulo de calor em seu interior. O corpo assim modelado encontra-se bem próximo à condição de neutralidade térmica [1]. A Tabela 2.1 detalha a proporção de trocas realizadas pelo organismo.

Tabela 2.1: Trocas térmicas

Processo	Frequência	Fenômeno
Radiação	40%	Emissão de raios infravermelho
Convecção	30%	Fluxo de ar quente expirado
Evaporação	20%	Calor latente de vaporização da umidade na superfície da pele
Respiração	8%	Evaporação de parte da água contida no ar
	2%	Aquecimento de gases respiratórios
Condução	Desprezível	Contato com objetos frios

Com base em um estudo feito com 1296 estudantes, Fanger propôs o PMV (*predicted mean vote ou voto médio previsível*) e o PPD (*Predicted Percentage of Dissatisfied ou Percentagem Previsível*)

de *Insatisfeitos*)[5]. O índice PMV varia de -3 a +3, onde o menor valor (-3) corresponde a muito frio e o maior valor (+3) muito quente. O zero corresponde ao equilíbrio ideal em se tratando de conforto térmico; a Equação 2.1 denota a expressão para cálculo do PMV [9].

$$PMV = 0.303(\exp^{-0.036M} + 0.028)L \quad , \quad (2.1)$$

onde M é a taxa de produção metabólica e o L é a carga térmica atuante sobre o corpo do indivíduo. A partir do PMV é possível calcular o índice PPD que denota a quantidade estimada de pessoas insatisfeitas termicamente com o ambiente. Ele se baseia na percentagem de um grande grupo de pessoas que gostariam que o ambiente estivesse mais quente ou mais frio. A Equação 2.2 expressa o cálculo do PPD [1].

$$PPD = 100 - 95 \exp^{-0.03353PMV^4 - 0.2179PMV^2} \quad (2.2)$$

A Figura 2.2 mostra o gráfico que relaciona a percentagem de pessoas insatisfeitas (PPD) em função do voto médio estimado (PMV).

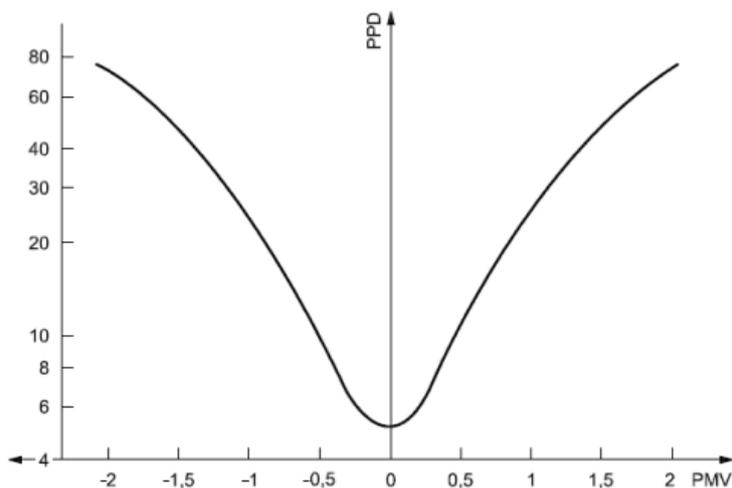


Figura 2.2: PPD *versus* PMV [1]

Devido às diferenças pessoais, não é possível ambientes projetados que agradem todos indivíduos. É plausível, entretanto, se especificar ambientes que sejam aceitáveis termicamente, ou seja, satisfaçam a maioria de seus ocupantes. A Tabela 2.2 proposta pela ASHRAE elucida temperaturas recomendáveis para o conforto térmico, embora não discrimine diferenças geográficas por exemplo.

Tabela 2.2: Padrão ASHRAE

	Temperatura operacional	Intervalo aceitável
Verão	22°C	20-23°C
Inverno	24,5°C	23-26°C

O intervalo aceitável presente na Tabela 2.2, de acordo com [1], especifica uma aceitabilidade

térmica de 90% dos presentes e um percentual de 85% dos ocupantes não estariam insatisfeitos com o ambiente. Dessa forma:

$$-0.5 < PMV < 0.5 \quad (2.3)$$

Dessa forma, deseja-se que o controle do ambiente (sala de reuniões do LARA) esteja dentro desta margem.

2.2 Propriedades psicométricas

Para um melhor entendimento da climatização é necessário um certo conhecimento sobre a composição do ar. O ar atmosférico é constituído de uma mistura de gases (N_2 , O_2 , CO_2 , etc), vapor de água e uma série de poluentes contaminantes como partículas sólidas em suspensão. Neste contexto, o ar seco é definido pela mistura de gases sem a presença de vapor d'água e os contaminantes [2].

De acordo com [14], o ar seco é composto por 78% de nitrogênio, 21% de oxigênio e 1% de dióxido de carbono, hélio, hidrogênio, néon e argônio. Esses valores costumam ser constantes, sendo assim o vapor d'água o componente variável do ar atmosférico, correspondendo de 1% a 3% da massa total. Dado que o vapor de água no ar resulta principalmente da evaporação de água da superfície de várias massas de água, como praias e lagos, a umidade atmosférica é maior em regiões localizadas perto dessas grandes massas de água e é menor nas regiões mais áridas [6].

Ocasionalmente, o índice de conforto térmico de um ambiente ou da atmosfera depende majoritariamente da quantidade de vapor de água presente no ar do que da própria temperatura em si. Assim sendo, um aparelho de ar condicionado pode ser projetado para atuar principalmente sobre a umidade em detrimento da temperatura, mesmo que a duas estejam consideravelmente relacionadas. Por consequência desses fatos e devido à crescente importância, o estudo detalhado da mistura de ar seco e vapor d'água passou a ser uma linha de estudo, a psicometria; que estuda as propriedades da mistura de ar seco e vapor d'água a partir de determinação de parâmetros específicos [2].

2.2.1 Pressões parciais

A relação de pressão parcial dos gases é governada pela Lei de Dalton. O inglês John Dalton propôs esta lei baseada em algumas conclusões importantes relacionadas à pressão parcial que cada gás exerce dentro de uma mistura a partir de várias observações de fenômenos meteorológicos e da composição do ar atmosférico.

A Lei de Dalton estabelece que em qualquer mistura de gases, cada gás presente na mistura exerce uma pressão parcial que corresponde a pressão total que o gás exerceria sozinho; além do mais, a pressão total de uma mistura de gases em um recipiente é igual à soma dos pressões parciais

dos gases que formam a mistura [6]. A expressão da Lei de Dalton é mostrada pela Equação 2.4.

$$P_{total} = \sum_{i=1}^n p_i \quad (2.4)$$

A Figura 2.3 ilustra a definição matemática dada.

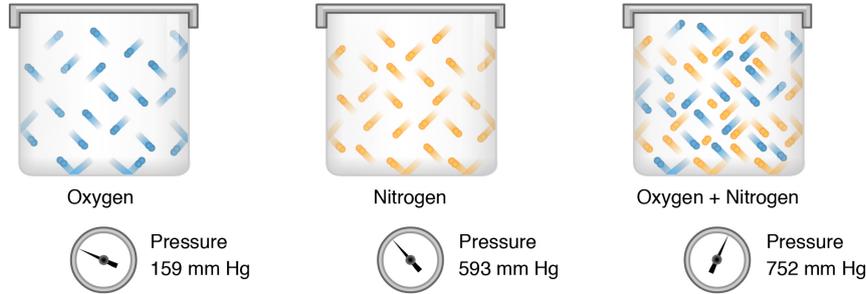


Figura 2.3: Lei de Dalton

Como visto na Seção 2.2, o ar atmosférico é uma mistura de gases com vapor d'água, logo deve obedecer à Lei de Dalton. A Equação 2.5 mostra a aplicação da lei para o caso do ar atmosférico, mistura que é a base da psicometria.

$$p_t = p_a + p_v \quad (2.5)$$

onde p_t é a pressão total, p_a é a pressão parcial do ar seco (soma das pressões parciais de nitrogênio, oxigênio, dióxido de carbono, etc) e p_v a pressão parcial do vapor d'água.

2.2.2 Umidade absoluta

Como sendo uma das variáveis que se almeja controlar no processo, é importante entender o conceito de umidade e as definições relacionada a tal. Neste contexto, pode-se definir a umidade absoluta como sendo a razão da massa de vapor d'água [kg] pela massa de ar seco [kg] em uma mistura vapor d'água-ar seco. Portanto:

$$w = \frac{m_v}{m_a} \quad (2.6)$$

A Equação 2.6 dá a expressão para a umidade absoluta. Sabendo que da expressão geral para a Lei do Geral dos Gases, pode-se obter a Equação 2.7:

$$n = \frac{PV}{RT} \quad (2.7)$$

como o número de moles é proporcional à massa, segue que a razão para a massa é a mesma. Logo:

$$w = \frac{m_v}{m_a} = \frac{R_a p_v}{R_v p_a} \quad (2.8)$$

onde a razão R_a por R_v é densidade relativa do vapor d'água com ar seco e depende das razão entre as massas moleculares de ambos; esta razão é de 0.622 [5]. Sabendo também que a pressão de

ar seco (p_a) é a pressão total (p_t) menos a pressão de vapor d'água (p_v), pode-se obter a expressão para umidade absoluta em função apenas p_v e p_t , como mostra a Equação 2.9.

$$w = 0.622 \frac{p_v}{p_t - p_v} \quad (2.9)$$

2.2.3 Umidade relativa

A umidade relativa do ar atmosférico é dada pela razão entre a pressão parcial do vapor d'água presente no ar sobre a pressão de saturação; ambos em relação à mesma temperatura. A pressão de saturação é definida como, a uma determinada temperatura, existe um percentual máximo de moléculas de água no ar em estado de vapor. Ultrapassado este percentual inicia-se a condensação do vapor de água. Se diz que o ar está saturado quando se atinge esse estado máximo de vapor d'água no ar.

$$\Phi = \frac{p_v}{p_{vs}} 100 \quad (2.10)$$

A Equação 2.10 mostra a expressão da umidade relativa que é dada em percentagem, com uma variação de 0%, quando o ar está completamente seco, a 100% onde o ar está úmido saturado, tipicamente condição de chuva.

Embora a umidade relativa não seja a variável diretamente controlada no processo, é um dos parâmetros de interesse do sistema visto que será medida e afetada no processo, assim como influente na temperatura. A tabela 2.3 mostra os parâmetros de conforto para umidade associados à temperatura.

Tabela 2.3: Parâmetros de conforto umidade [6]

	Temperatura	Umidade
Verão	22.5-25.5°C	65%
	23-26°C	35%
Inverno	21-23°C	60%
	21.5-24°C	30%

2.2.4 Temperatura de bulbo seco e temperatura de bulbo úmido

A tempera de bulbo seco (TBS) do ar atmosférico nada mais é do que a temperatura medida por um termômetro comum convencional. Já a temperatura de bulbo úmido (TBU) é a temperatura que reflete as propriedades psicométricas de um sistema constituído pela evaporação da água do ar. A temperatura de bulbo úmido é a temperatura mais baixa que pode ser alcançada apenas pela evaporação da água. Ao contrário da temperatura de bulbo seco, a temperatura de bulbo úmido é uma indicação da quantidade de umidade no ar. Quanto menor a umidade relativa do ar, maior o resfriamento.

Para se mensurar a BTU utiliza-se um termômetro comum com o seu bulbo envolvido por uma mecha umedecida. Para que ocorra a medição deve-se expor o termômetro à uma corrente de

ar. A temperatura de bulbo úmido cai devido ao calor retirado para evaporar a água; ou seja, o resfriamento é diretamente proporcional à secura do ar.

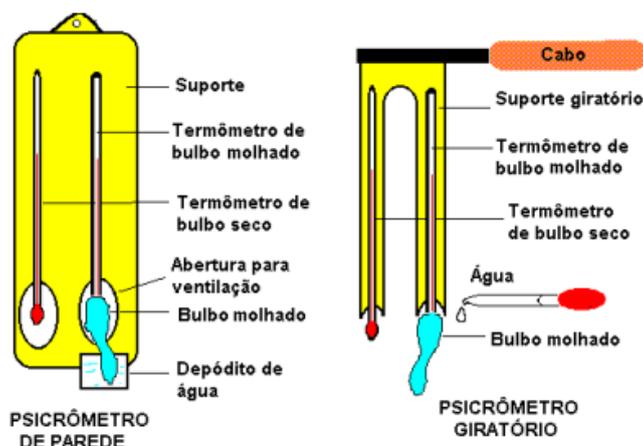


Figura 2.4: Psicômetro [2]

A Figura 2.4 representa um psicômetro, instrumento utilizado para medir a TBU. A temperatura registrada por um termômetro TBU será sempre mais baixa que a temperatura TBS do ar, com exceção do caso onde o ar está saturado, nesta ocasião os termômetros mostram o mesmo valor. A diferença entre a TBS e TBU é chamada depressão de bulbo úmido, quanto maior essa diferença, menor a umidade relativa do ar.

2.2.5 Temperatura de orvalho

Temperatura de ponto de orvalho é a temperatura em que o ar se encontra saturado de vapor d'água. Logo, quando o ar atinge o ponto de orvalho para uma dada pressão, o vapor d'água está em equilíbrio com a quantidade de água líquida do ambiente; ou seja, o vapor d'água condensa à uma mesma taxa em que a água líquida evapora.

Portanto, neste contexto, com uma temperatura abaixo da temperatura de orvalho a água no ambiente condensará, principalmente ao redor de superfícies sólidas. Temperatura de ponto de orvalho está diretamente ligada à umidade relativa do ar e é comumente utilizada como um meio mais preciso pra medir a umidade visto que é uma grandeza absoluta.

2.3 Carta psicométrica

As cartas psicométricas são representações das propriedades psicométricas do ar [14]. O uso destas cartas, ou diagramas, permite de maneira prática uma análise gráfica de dados e processos psicométricos. Desta forma, o diagrama por si só funciona como uma tabela que relaciona valores (umidade absoluta, relativa, TBU, TBS, etc) facilitando bastante a resolução de problemas. A Figura 2.5 representa uma carta psicométrica típica.

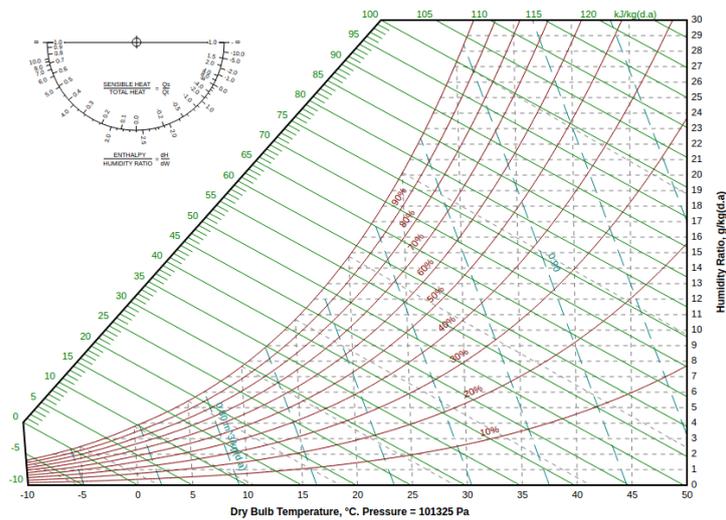


Figura 2.5: Carta psicrométrica (flycarpet.net/en/psyonline)

Da carta, é possível se obter as seguintes propriedades conforme a análise das intersecções das linhas correspondentes. [6].

- | | | |
|----|-----------------------------------|-------------------------------------|
| 1. | Temperatura de bulbo seco (TBS) | [°C] |
| 2. | Umidade absoluta (w) | [kg _v /kg _a] |
| 3. | Escala de umidade absoluta | |
| 4. | Temperatura de bulbo úmido (TBU) | [°C] |
| 5. | Volume específico (v) | [m ³ /kg _a] |
| 6. | Escala de entalpia específica (h) | [kJ/kg _a] |
| 7. | Temperatura de ponto de orvalho | [°C] |
| 8. | Escala calor sensível | |

Desta forma, as linhas retas verticais representam a TBS; as linhas retas horizontais representam a umidade absoluta; a linha da extrema esquerda curvada representa a escala da TBU. A Figura 2.6 representa estas principais propriedades na carta psicrométrica conforme os índices.

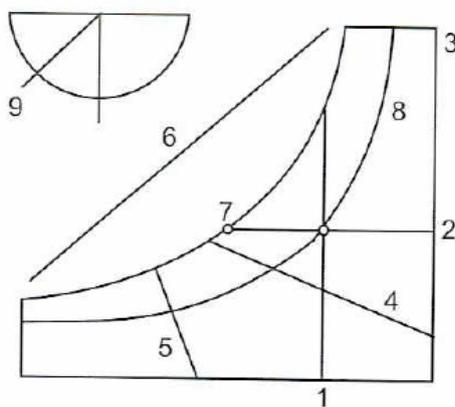


Figura 2.6: Propriedades da carta psicrométrica

Os processos psicrométricos podem ser ilustrados e analisados num diagrama psicrométrico. Pode-se definir estes processos psicrométricos como as transformações ocorridas nas propriedades do ar durante o processo de climatização, como por exemplo umidificação, resfriamento e aquecimento [6].

Mesmo a carta psicrométrica, que facilita bastante na maioria dos casos, pode ser de difícil implementação em sistemas de tempo real como o processo que necessita ser controlado. Uma representação simplificada da dinâmica do processo de climatização como o proposto por Pedro Ivo [5] requer a estimação da TBU e da umidade absoluta.

Neste contexto, Stull [15] propôs empiricamente uma equação que calcula a temperatura de bulbo úmido a partir da temperatura de bulbo seco e da umidade relativa do ar. A Equação 2.11 mostra a expressão proposta por Stull.

$$T_w = T \arctan[0.151977(RH\% + 8.313659)^{\frac{1}{2}}] + \arctan(T + RH\%) - \arctan(RH\% - 1.676331) + 0.00391838(RH\%)^{\frac{3}{2}} \arctan((0.023101RH\%) - 4.686035) \quad (2.11)$$

onde T_w é a temperatura de bulbo úmido, T a temperatura de bulbo seco e $RH\%$ a umidade relativa do ar. A expressão é válida para pressão de 101.325 kPa, corresponde a pressão à nível do mar, com uma erro absoluto de 0.3°C para temperaturas entre -20° e 50°C.

Capítulo 3

Automação do sistema de climatização híbrida

Este capítulo aborda o sistema de climatização híbrida, assim como a automação do processo

3.1 Sistema de climatização híbrido do LARA

O sistema de climatização híbrido presente na sala de reuniões do LARA opera em dois modos principais, convencional e evaporativo, e tem finalidade de propiciar melhor qualidade de ar interior visando a eficiência energética e conforto térmico que se mostre vantajoso em relação ao sistema convencional.

O sistema presente no LARA foi construído a partir da adaptação de um sistema *split*, do qual foram aproveitados os componentes e foi efetuado um novo projeto de controle eletrônico. A parte evaporativa foi construída utilizando um painel modelo Celdek da Munters e uma bomba de água. Além disso, foi instalado um *dampner* para a renovação e retorno parcial do ar [6], neste caso em especial para o modo convencional.

3.1.1 Funcionamento convencional

O funcionamento de um ar condicionado convencional é um ciclo de refrigeração fechado onde circula um fluido refrigerante a fim de manter continuamente sua evaporação, baseado na compressão. O ciclo de refrigeração é composto, de forma simplificada, basicamente de um compressor, um condensador, uma válvula de expansão e um evaporador. A Figura 3.1 ilustra a estrutura básica de um ciclo de refrigeração.

Pode-se analisar o fluxo começando pelo compressor que succiona o fluido refrigerante à baixa pressão e baixa temperatura e o comprime, elevando assim sua temperatura e pressão. Ao chegar à válvula de expansão, o fluido que na fase de compressão passou de gás para líquido perde pressão e baixa sua temperatura.

O fluido, agora à baixa pressão e temperatura, passa pelo evaporador e muda novamente de fase ao retirar calor do ambiente que se deseja resfriar, tornando-se vapor à baixa pressão e temperatura.

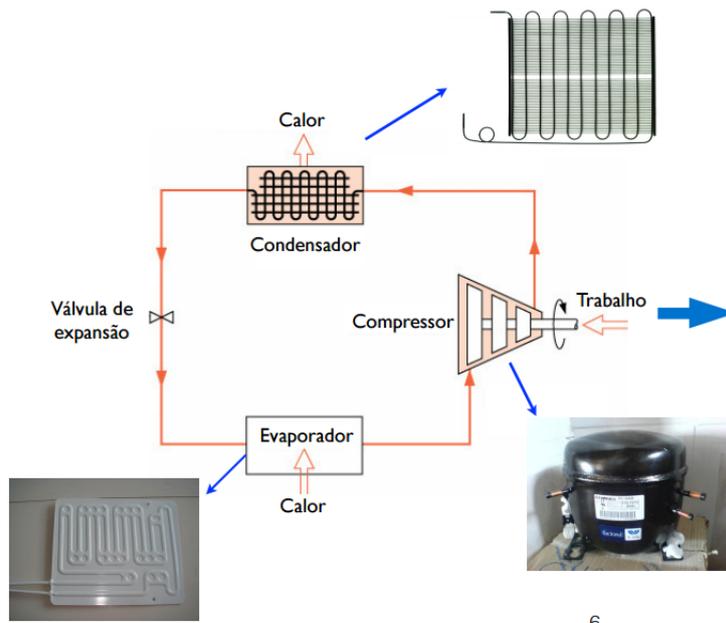


Figura 3.1: Ciclo de refrigeração [3]

Este vapor volta a ser succionado pelo compressor, repetindo-se o ciclo. O compressor é o principal responsável pelo alto consumo de energia elétrica do ar condicionado convencional

3.1.2 Funcionamento evaporativo

O condicionamento de ar por resfriamento evaporativo é um método ambientalmente amigável e energeticamente eficiente, que utiliza água como fluido de trabalho e pode ser uma alternativa econômica aos sistemas convencionais de ar condicionado em muitos casos [4].

O resfriamento evaporativo faz uso dos fenômenos que ocorrem quando água e ar úmido insaturado entram em contato. Calor é transferido do fluido de maior temperatura para o de menor temperatura, e vapor d'água vai da região de maior pressão de vapor para a região de menor pressão.

O processo de resfriamento só é possível graças à interação que há entre a transferência de calor e a transferência de massa. A água absorve calor do ar na evaporação e calor é retirado do vapor d'água na condensação. O contato livre e prolongado entre a água e o ar resulta num estado de equilíbrio em que normalmente o ar estará saturado. Quanto maior a temperatura do ar, maior é a quantidade de vapor d'água no ar necessária para saturá-lo.

O processo de resfriamento evaporativo direto ocorre quando água líquida entra em contato com o ar de alimentação a ser inserida no ambiente a ser climatizado, contato feito por uma manta de celulose. O calor sensível presente no ar de alimentação evapora a água líquida, e como resultado tem-se a redução da temperatura do ar e aumento da sua umidade. O processo de evaporação continua até que o ar esteja saturado e a mistura ar vapor entre em equilíbrio.

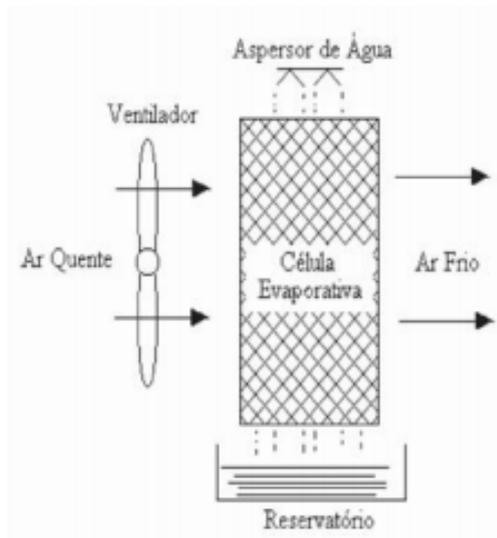


Figura 3.2: Sistema evaporativo [4]

A Figura 3.2 mostra o sistema evaporativo. Idealmente o processo é adiabático, ou seja, nenhum calor entra ou sai do sistema, e ocorre somente quando a água de resfriamento está na temperatura de bulbo úmido (TBU) do ar.

Para o caso do sistema do LARA, que é nosso objeto de estudo, o resfriador de ar possui um ventilador que aspira ar externo através de um painel evaporativo, sobre o qual água é circulada continuamente por uma pequena bomba. A água que evapora é repostada por uma boia que mantém nível constante no reservatório.

O painel evaporativo do resfriador de ar normalmente é composto por camadas de papel *kraft* de alta qualidade, impregnado com uma resina que lhe confere grande rigidez e durabilidade. Uma vez coladas, as camadas formam blocos de área superficial muito grande, que oferecem baixa resistência ao fluxo de ar.

Portanto, para que haja esta absorção é necessário que a água utilizada passe da fase líquida para a fase vapor. Esta mudança de fase demanda uma quantidade de energia que é retirada do meio, no caso o ar, resfriando-o. Por um princípio básico nas reações físico-químicas segundo o qual quanto maior a superfície de contato entre os reagentes, maior a velocidade da reação. Assim sendo, devemos procurar aumentar a área de contato entre a água e o ar. Como o ar já se encontra diluído e ocupando todos os espaços disponíveis, resta a água para dispersar, esse é o papel da manta de celulose no processo.

Do processo conclui-se que a diminuição da temperatura será proporcional à umidade relativa do ar captado, quanto mais seco maior será a redução. A temperatura do ar resfriado tende a ficar próxima à TBU do ar captado, em torno de 1°C acima. Como a temperatura externa é inversa a umidade relativa, em dias mais quentes e secos o desempenho do sistema será melhor.

Por fim, mas não menos importante, vale ressaltar também que a vazão de ar no sistema evaporativo é relevante para o desempenho do mesmo; porém, isso pode levar a um nível de ruído incompatível com o ambiente que se deseja climatizar. Dessa forma, o modo evaporativo será

utilizados com foco em dar suporte ao modo convencional e para um controle melhor da umidade

3.2 openHAB

O openHAB é um sistema supervisorio, do inglês *Open Home Automation Bus*, que se caracteriza por ser um software livre de automação tendo como principal função a integração dos diferentes sistemas e suas respectivas tecnologias que formam uma solução inteligente e elegante, permitindo assim uma implementação robusta e extensiva de regras de automação integrada a uma interface gráfica amigável; razões pelas quais foi utilizado neste trabalho.

Muito presente desde o advento da Internet das Coisas, o openHAB é largamente utilizado em projetos de automação especialmente em aplicações voltadas para o IoT. Neste contexto, trata-se de um software agnóstico tanto com relação ao *hardware* quanto em relação ao protocolo usado, ou seja, pode rodar em diferentes plataformas e com a capacidade de estabelecer conexão com uma grande variedade de protocolos como o usuário desejar.

Neste trabalho foi utilizado sua versão *runtime* que é escrita em Java e com arquitetura de modularização OSGi. Por característica, a versão *runtime* fornece comandos de baixo e de alto nível de um modelo de execução de uma linguagem de programação, ou seja, fazendo a interface entre o software e o conjunto de instruções da arquitetura do hardware [8]. As funções de baixo nível fornecidas pelo sistema *runtime* incluem, por exemplo, o carregamento da memória, execução paralela de softwares, a conversão digital para binário, a correta manipulação da pilha do processador após uma chamada de função, entre outras.

O openHAB trabalha com uma implementação orientada a eventos, onde a comunicação é tratada de duas formas diferentes pelos canais internos: um barramento assíncrono de eventos e um repositório de estados. Assim sendo, comando disparados pelo usuário ou de maneira programada resultam em uma ação ou mudança de estado de um item ou serviço. Da mesma forma podem ocorrer atualizações, provenientes de um Arduino por exemplo, que fornecem informações sobre o estado de mudança de um item ou serviço.

Como a maioria dos sistemas supervisorios, o openHAB trabalha como um ponto central de integração que atua como mediador entre os diferentes protocolos implementados por diversos dispositivos a ele conectados. O sistema utiliza então um repositório de estados responsável por fornecer e atuar sempre o usuário ou lógica do programa requisitar e respeito de item qualquer. A Figura 3.3 mostra um exemplo de item representado pela lâmpada onde por meio do supervisorio pode-se monitorar seu estado, assim como enviar comandos para controle. Os itens podem assumir diferentes formas, como *switches* e *dimmers*.

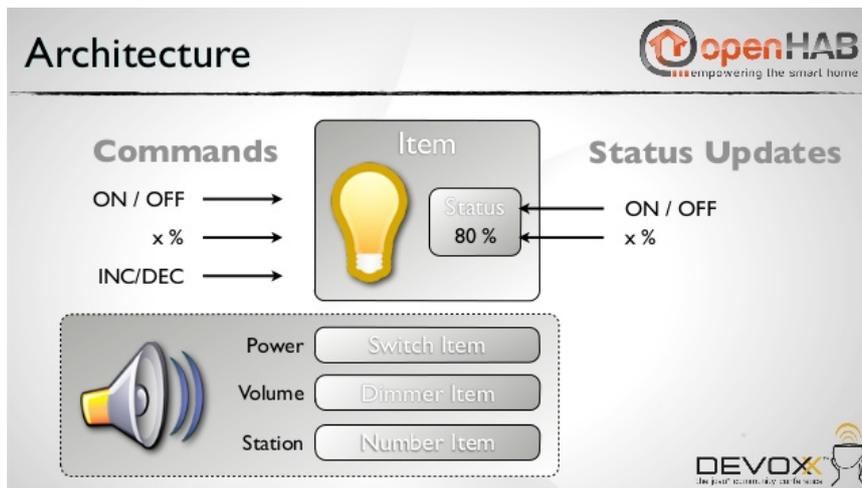


Figura 3.3: Esquemático openHAB

Portanto, o openHAB será responsável por fazer o monitoramento de todos os estados que se fazem necessários, como temperatura e umidade da sala de reuniões assim como temperatura externa e da sala vizinha e enviar comando caso necessário. Outros itens importantes também são as informações com relação aos atuadores do sistema sendo ele o compressor, a bomba d'água e o *damper*. A Figura 3.4 mostra a interface gráfica mobile para o sistema de climatização híbrida com todos os itens que são relevantes para o usuário.

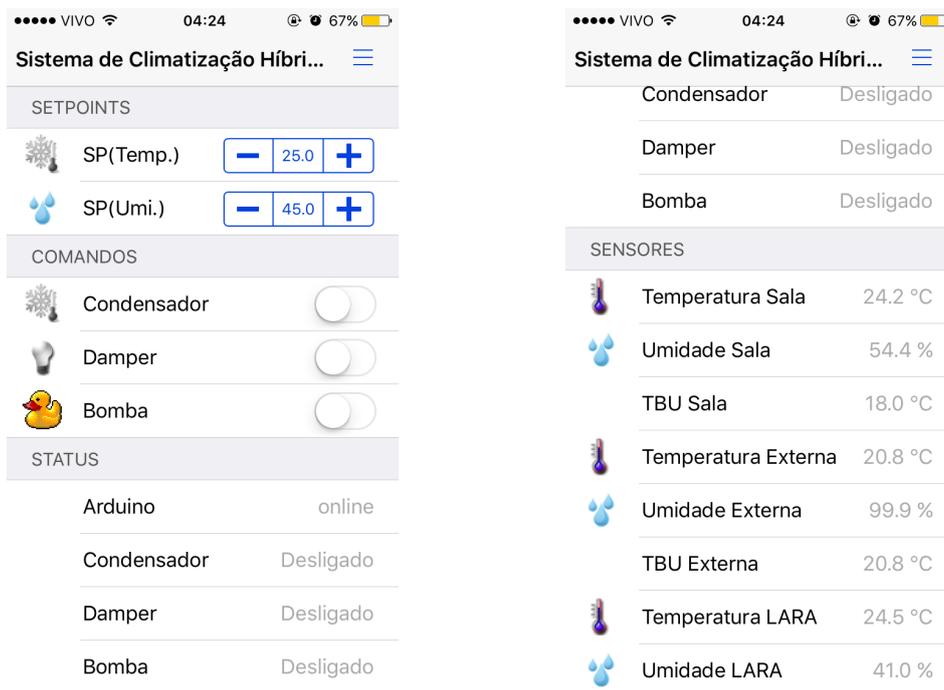


Figura 3.4: GUI openHAB

O supervisor em si já possui um serviço de persistência nativo para o armazenamento dos dados, o *rrd4j*. Como complemento para uma melhor persistência registros dos itens, foi instalado um servidor MySQL database para uma melhor manipulação dos dados pois é amplamente

utilizado, livre e de fácil integração com o Matlab por exemplo para a análise dos resultados.

Por fim, vale ressaltar que as funcionalidades do programa crescem diariamente visto se pode instalar pacotes adicionais feitos por terceiros. Por ser um software livre, a comunidade de colaboradores tem facilidade em estender esses pacotes adicionais que são conhecidos com *bindings*.

Para que se pudesse atuar no sistema de forma remota, não sendo necessário estar fisicamente no Lara, foi conectado o sistema supervisor à internet e implementado o serviço web *my.openhab*. Desta forma é possível acessar o sistema de qualquer lugar, com todos os dados da persistência salvos em nuvem.

3.3 MQTT

Do inglês *Message Queue Telemetry Transport*, MQTT é um protocolo de comunicação do tipo M2M (*machine-to-machine*) voltado para internet das coisas. Com sua troca de mensagens fundamentada no modelo do tipo publicador/subscritor, é extremamente leve e simples, projetado especialmente para dispositivos restritos e de baixa largura de banda, alta latência e otimizado para redes TCP/IP não confiáveis. Utilizado principalmente em telemetria, seu princípio de design é minimizar a largura de banda da rede e os requisitos de recursos do dispositivo enquanto também tenta garantir a confiabilidade e algum grau de garantia de entrega.

Este modelo, que é o mesmo de várias aplicações web, define uma dependência um-para-muitos entre instâncias de modo que quando um item altera seu estado ou é alterado, todos os objetos à ele subscritos são notificados e atualizados automaticamente. Assim sendo, qualquer objeto interessado em determinado tópico seja avisado da mudança de estado de outros objetos ou eventos que estão ocorrendo; se o objeto estiver ouvindo o canal, ou seja subscrito no tópico.

O principal elemento da comunicação no protocolo MQTT é o *message broker*, que é o módulo principal responsável por corrigir a mensagem; ou seja, é um programa que traduz uma dada mensagem de um protocolo do remetente para uma mensagem no protocolo do destinatário. O *broker* de mensagens mais usado e que também foi utilizado neste trabalho foi o mosquitto, que foi rodado na unidade principal juntamente com o openHAB. A Tabela 3.1 mostra os principais métodos utilizados por um objeto que se conecta ao *broker*.

Tabela 3.1: Métodos do MQTT

Método	Ação
<i>Connect</i>	Estabelecer conexão entre servidor/broker
<i>Disconnect</i>	Finalizar a conexão TCP/IP após término da ação
<i>Subscribe</i>	Realiza subscrição de um cliente em determinado tópico
<i>Unsubscribe</i>	Termina a subscrição de um cliente em determinado tópico
<i>Publish</i>	Publica uma mensagem em determinado tópico. Todos os subscritos podem receber a mensagem

É importante ressaltar que pelo processamento das mensagens por meio de tópicos não é possível um remetente enviar uma mensagem diretamente a um destinatário, mas sim ao *broker*, diferente do que ocorre na maioria dos protocolos de comunicação utilizados; ou seja, não se estabelece uma conexão direta *p2p*. O *broker* recebe a mensagem do publicador e faz uma filtragem, redirecionando para o tópico de destino.

Os tópicos podem ser organizados de forma hierárquica em vários níveis, como numa estrutura tipo árvore. Pode-se criar por exemplo o tópico Lara, dentro deste tópico criar dois sub-tópicos 'reuniao' e 'are_comum'; dentro destes pode-se criar ainda mais sub-tópicos 'temperatura' e 'umidade'. Desta forma, uma entidade que deseje saber unicamente a temperatura da sala de reuniões irá se inscrever no tópico 'Lara/reuniao/temperatura'; da mesma forma alguém que deseje saber todas as umidades de todas as salas do LARA pode se inscrever no tópico 'Lara/#/umidade'; ainda mais, se for interesse saber de todos os dados produzidos pelo Lara pode-se inscrever no tópico 'Lara/#'. A Figura 3.5 exemplifica a estrutura do MQTT.

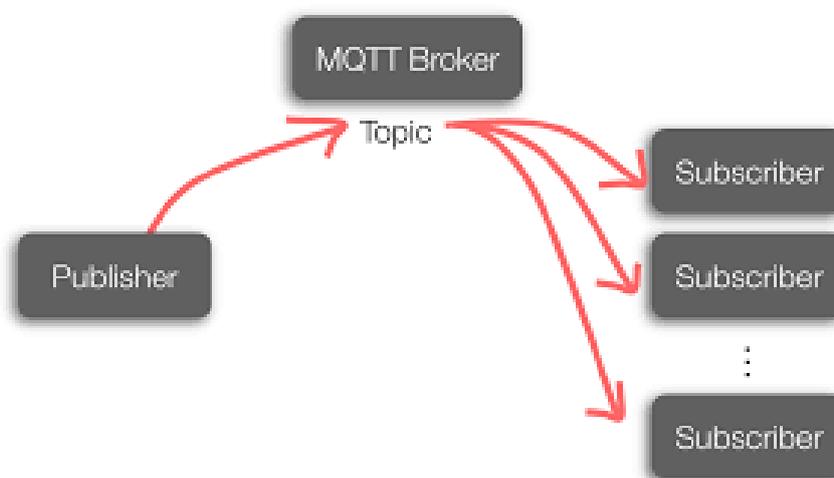


Figura 3.5: Protocolo MQTT

Esta abordagem tem como desvantagem a segurança, visto que a princípio qualquer entidade pode 'ouvir' mensagens em qualquer tópico assim como publicar mensagens que não são interessantes. Para contornar tal problema, é possível criar tópicos com chaves onde só quem possui a combinação pode publicar ou subscrever. Neste trabalho não foi necessário tal abordagem pois se implementou o *broker* em uma rede interna do Lara com senha.

3.4 Materiais

3.4.1 Sensor DHT22

O DHT22 é um sensor digital de temperatura e umidade simples e relativamente de baixo custo. Se utiliza de um sensor de umidade capacitiva e um termistor para medir o ar que circula envolta, gerando um sinal digital no pino de saída. É bastante simples de usar, mas possui certa limitação

com relação ao tempo de amostra do dados a serem capturados. Tal desvantagem restringe o sistema à amostras de com intervalos de 2 segundos.



Figura 3.6: Sensor DHT22 (adafruit.com)

O DHT22 tem sua temperatura compensada e calibrada em uma câmara de calibração e seu coeficiente de calibração é salvo em uma memória OTP (One-Time Programmable). Toda vez que o sensor é lido, ele irá ler esse coeficiente de sua memória. Alguns detalhes técnicos são:

- Baixo custo
- 3 a 5V de alimentação, compatível com o Arduino
- Range de umidade 0 - 100% com acurácia de 2 - 5%
- Range de temperatura -40 a 80°C com acurácia $\pm 0.5^{\circ}\text{C}$

Como dito o sensor vem calibrado de fábrica, sem necessidade de posterior calibração. Um novo sensor foi comprado para se substituir e foi utilizado para comparar com os outros, afim de checar sua calibração. A Tabela 3.2 mostra os pinos do sensor DHT22 e sua função; alguns podem vir só com 3 pinos com um circuito que evita saturação devido à taxa de amostragem.

Tabela 3.2: Pinagem do DHT22

Pino	Função	
1	VCC	- tensão de alimentação
2	DATA	- Serial de dados
3	NULL	- Não utilizado
4	GND	- Referência

Algumas de suas principais vantagens são o tamanho pequeno, o baixo consumo próprio para sistemas integrados com Arduino e boa distância de transmissão de dados (20m). No trabalho foram usados três sensores: dentro da sala de reuniões, externo fora do SG11 e um no Lara fora da sala de reuniões. Mais especificações técnicas do sensor DHT22 estão presentes no Anexo.

3.4.2 Raspberry Pi

O Raspberry Pi é uma série de computadores que se caracteriza pelo seu tamanho reduzido, comparado a um cartão de crédito. Foi desenvolvido no Reino Unido pela Fundação Raspberry Pi para a princípio promover o ensino da informática básica em escolas e países em desenvolvimento.



Figura 3.7: Raspberry Pi (raspberrypi.org)

Atualmente o Raspberry é largamente utilizado em sistemas de automação e aplicações embarcadas devido ao seu tamanho reduzido e alto poder de processamento aliados ao baixo consumo de energia. Seu sistema operacional mais comum é uma distro Linux baseada no Debian. Para esta trabalho foi utilizado o Raspberry Pi 2 Model B, algumas de suas especificações são:

- Chip: Broadcom BCM2835 SoC
- Processador: ARM Quad-Core Cortex-7
- CPU: 900 MHz
- Tensão de operação: Micro USB Socket 5V/2A
- 4 portas USB
- Full HDMI
- Porta Ethernet 10/100 (RJ45)
- Slot Micro SD Card

O RPi, como é conhecido o Raspberry Pi, será neste projeto o ponto central do sistema de automação. Nele foram instalados o software supervisor openHAB, o *broker* MQTT responsável pela transmissão dos dados entre o Arduino e o supervisor assim como estados e comandos necessários. Também foi instalado, como já dito, o banco de dados MySQL e o *Dataplicity* que se trata de uma aplicação web que permite o acesso remoto à linha de comando do RPi.

3.4.3 Arduino

Arduino é uma placa de prototipagem eletrônica de hardware livre e de placa única, projetada com um microcontrolador Atmel AVR e uma IDE para desenvolvimento de software. O hardware da placa dá suporte a entrada e saída em diversas formas de dados de maneira simples, com a quantidade e capacidade de processamento dependendo do modelo. O ambiente de desenvolvimento do software é voltado para uma linguagem padrão própria baseada no Wiring, uma variação do C/C++. Além do compilador, há também o *bootloader* - programa grava no chip que permite a carregamento de programas [9].



Figura 3.8: Arduino Mega (arduino.cc)

A Figura 3.8 mostra o Arduino Mega 2560, que possui o microcontrolador ATmega 2560. Esta foi a placa escolhida para este trabalho por possuir uma maior capacidade de processamento e memória, além de mais pinos de entrada e saída se comparada com outras placas mais comuns. A Tabela 3.3 mostra algumas especificações da placa.

Tabela 3.3: Especificações do Arduino Mega 2560

	Arduino Mega
Tensão de operação	5V
Tensão de entrada	7-12V
Pinos digitais	54, onde 14 podem ter saída PWM
Entradas analógicas	16
Corrente DC pino	20 mA
SRAM	8 kB
EEPROM	4 kB
Memória Flash	256 kB

possui ainda 4 UARTs, um cristal de 16 MHz, conexão serial via USB, conector de força, um conector ICSP e um botão *reset*.

Neste trabalho o Arduino foi utilizado como dispositivo de entrada e saída de dados dos sensores, interface de comando para acionamento dos atuadores e como ponto central da implementação do algoritmo de controle. O DHT22 manda para o Arduino, de forma cabeada, a TBS e umidade, que por sua vez manda pela rede através do MQTT para o openHAB. O openHAB funciona como um observador para monitoramento dos dados, além de interface para o *setpoint* de umidade e

temperatura desejados pelo usuário. A partir daí, com a referência e os dados dos sensores o Arduino implementa o controle e atua sobre as cargas, que no caso são relés.

3.4.4 Módulo Ethernet ENC28J60

Este modelo, o ATmega 2560, não possui porta ethernet. Logo, para se estabelecer uma conexão com a rede e conseqüentemente com o openHAB, é necessário o uso de um *shield* ou módulo que funciona como dispositivo para estender as funcionalidades do Arduino. O módulo Ethernet ENC28J60 é uma solução simples e barata porém eficiente que cumpre o papel de integrar à placa ao supervisor embarcado no RPi.



Figura 3.9: Módulo Ethernet ENC28J60 (blog.filipeflop.com/)

O módulo, ilustrado pela Figura 3.9, é composto pela porta ethernet RJ45, o chip controlador ENC28J60, um cristal de 15MHz e um conector de 10 pinos. A alimentação é 3.3V e a ligação do módulo é feita via interface SPI. A conexão entre o módulo e o Arduino foi feita conforme a Tabela 3.4. Para o objetivo desta implementação os pinos CLKOUT e WOL não são necessários.

Tabela 3.4: Conexão Módulo

Módulo ENC28J60	Arduino Mega
CS	D53
SI	D51
SO	D50
SCK	D52
RESET	RESET
INT	D2
VCC	3V3
GND	GND

A instalação de uma biblioteca particular para o ENC28j60 se fez necessária visto que esse módulo não está no mesmo padrão do *shield* ethernet para Arduino. O módulo foi então conectado via porta ethernet à um roteador que por sua vez está conectado ao Raspberry. Com todas as instâncias devidamente conectadas em rede, foram arbitrados endereços de IP fixo para os elementos afim de garantir a funcionalidade dos códigos¹, sendo 192.168.1.4 para o RPi e 192.128.1.101 para

¹Os códigos se encontram no anexo

o Arduino.

3.4.5 Módulo Relé

Um relé é um dispositivo comutador eletromecânico, ou seja, um interruptor. Seu funcionamento se dá quando uma corrente circula pela bobina de suas espiras que criam um campo magnético que atrai um ou uma série de contatos, fechando ou abrindo circuitos. De maneira análoga, quando se interrompe a corrente o campo magnético também é interrompido, resultando na volta dos contatos ao estado inicial.

A principal vantagem do relé é que o circuito de carga se encontra completamente isolado do de controle, o que trás como benefício a possibilidade de trabalhar com tensões diferentes entre a unidade de controle e carga. A desvantagem é que, por ser um componente mecânico, há um desgaste natural conforme o uso reduzindo assim o tempo de vida.



Figura 3.10: Módulo com 4 relés (Modelo: SRD-05VDC0SL-C)

Portanto, conforme os comandos de controle provenientes dos pinos digitais do Arduino acionam o relés, estes são responsáveis por acionar as contatoras que fazem o contato do circuito de potência do compressor, da bomba d'água e do *damp*er.

O módulo, conforme a Figura 3.10, já vem com o circuito de isolamento entre o controle e o acionamento. Algumas de suas características são: tensão de operação de $5 V_{cc}$; controla cargas de até $220 V_{ac}$; corrente de controle de $15\sim 20$ mA; corrente máxima de acionamento de 10A; tempo de resposta de $5\sim 10$ ms; estados NA (normalmente aberto) e NF (normalmente fechado). Foi feita uma otimização¹ em relação ao trabalho de [8], tanto na parte de software quanto de hardware.

¹A Figura 1 do Anexo mostra as conexões e pinagem dos componentes

3.5 PWM

Como visto na seção anterior, o relé só pode assumir dois estados: conduzir e não conduzindo corrente que resultam em estado de ligado ou desligado de certo equipamento. Dessa forma, é necessário traduzir o sinal de controle analógico gerado pelo controlador em um sinal digital para o controle do relé.

Neste contexto que se aplica o PWM, que é o acrônimo para *pulse width modulation*, ou modulação por largura de pulso em português. Consiste em uma técnica de modulação em que a largura do pulso é adequada ao modulador de sinal de informação [9]. Usado largamente tanto no controle de fornecimento de potência a dispositivos elétricos quanto em sistemas de comunicação, o PWM tem como vantagem a baixa perda de potência durante a transmissão.

O valor da tensão no pino de controle do relé que alimenta a carga varia com um pulso de valor 0 (LOW) a 1 (HIGH). A razão entre o tempo em que o pulso está em 1 e o período total do pulso é denominada ciclo de trabalho. Esta razão determina o quanto de potência média é entregue para a carga. Estima-se que o compressor por exemplo deva ter um período de aproximadamente 4 minutos entre um chaveamento consecutivo; logo, para uma potência média 75% por exemplo deve se manter o sinal em 1 durante 3 minutos e em 0 o minuto restante.

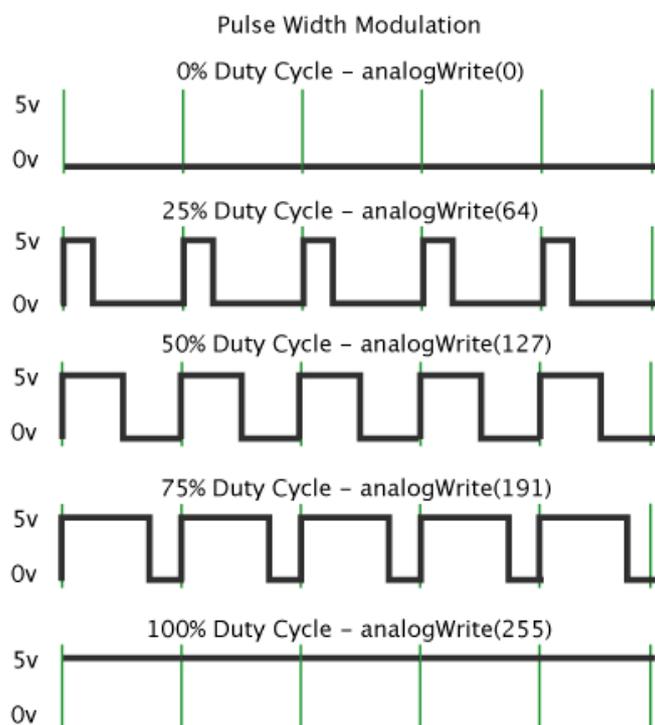


Figura 3.11: Exemplo de PWM aplicado ao Arduino

A Figura 3.11 mostra alguns exemplos de trens de pulso do tipo PWM gerados a partir de um valor associado à porta analógica do Arduino. Percebe-se que o período é constante, enquanto que a relação entre sinal em 0 e 1 que pode variar.

A Figura 3.12 mostra o diagrama de conexão dos componentes

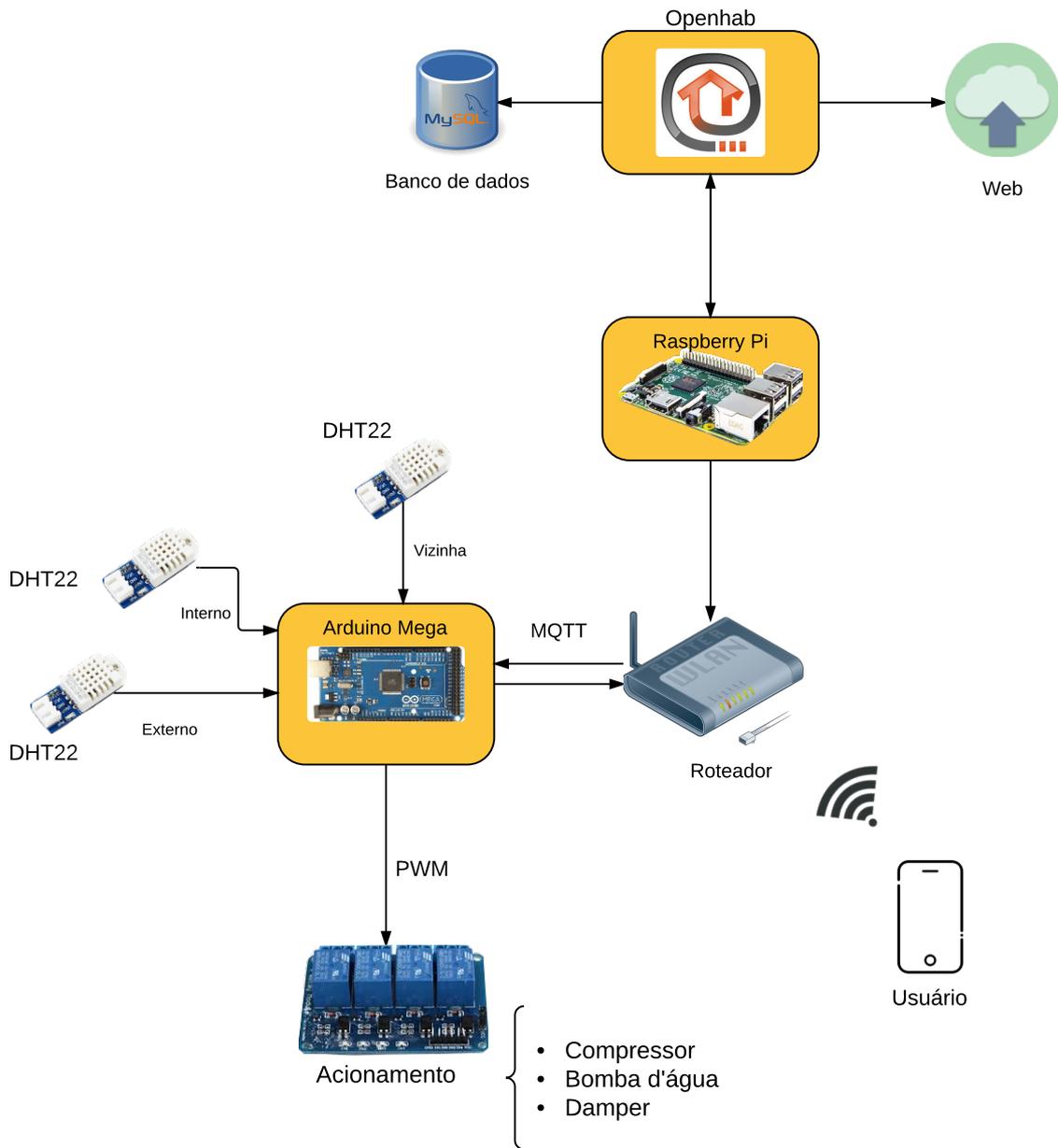


Figura 3.12: Diagrama completo de conexão

Capítulo 4

Controle TITO

Este capítulo aborda o projeto, desenvolvimento e implementação da estratégia de controle do sistema híbrido de climatização.

4.1 Modelo do Sistema

Baseado no trabalho de [5] que fez a identificação de processo, pode-se criar um modelo de sistema para simulações e entendimento do processo de climatização tanto no modo convencional quanto no modo evaporativo. A Figura 4.1 mostra a sala que será o ambiente de estudo desde trabalho junto com o esquema de climatização híbrido.

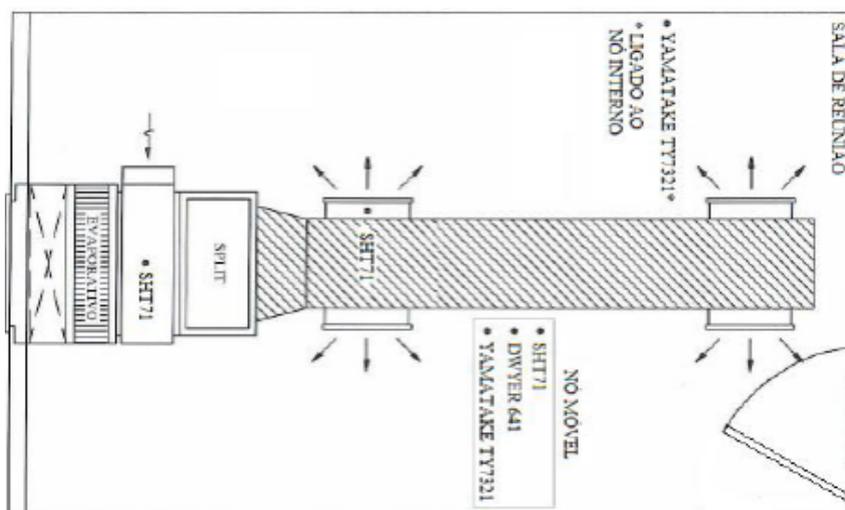


Figura 4.1: Sala de reuniões do LARA [5]

Pode-se dividir o sistema em dois, a dinâmica do processo referente à temperatura e a dinâmica referente à umidade. Analisando primeiro a malha referente à temperatura sabe-se que, embora seja a temperatura a variável a ser controlada, pelo modelo princípios fundamentais trabalha-se com o fluxo de calor no ambiente. Dessa forma, deve-se levar em consideração a influência na dinâmica do sistema, além da própria sala, a dinâmica da sala vizinha e do ambiente externo ao laboratório. Soma-se à este esquema o fluxo de calor referente à radiação solar.

Neste contexto, com o auxílio do Matlab, pode-se fazer um modelo de diagrama de blocos da dinâmica do processo no Simulink, em um primeiro momento com foco na temperatura e no modelo convencional, baseado em [5]. Levando em conta apenas a malha da temperatura, a Figura 4.2 representa o diagrama de blocos do processo.

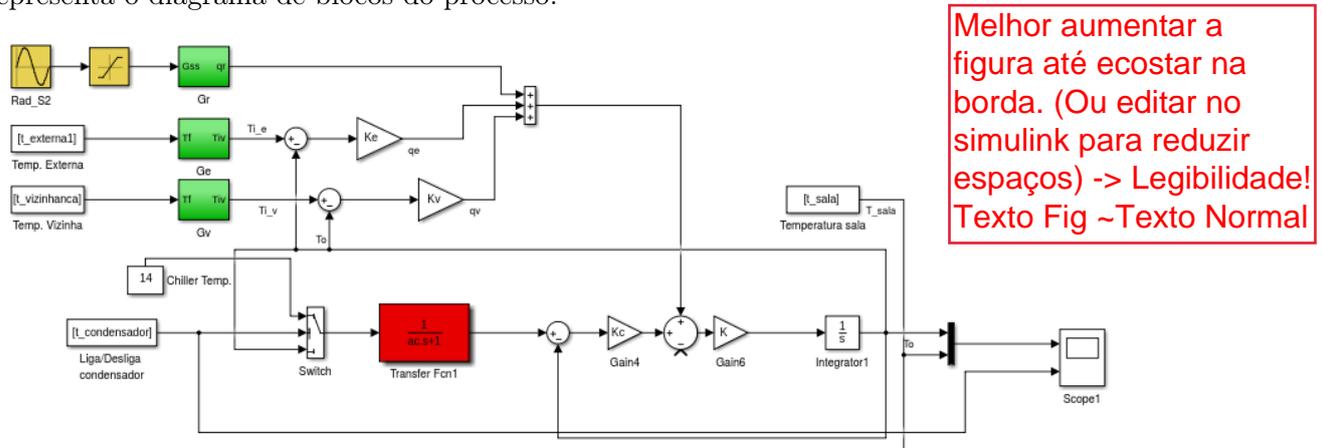


Figura 4.2: Modelo convencional

Da figura, tem-se que o bloco em vermelho representa a função de transferência do ar condicionado, onde a_c representa a constante de tempo do mesmo; ou seja, ~~o tempo necessário para que se note a mudança de temperatura.~~ Os blocos em verde G_r , G_e e G_v representam as funções de transferência da radiação, da temperatura externa e da sala vizinha respectivamente. Essas três componentes externas à sala são tratadas como perturbações.

Sabe-se que o processo térmico funciona à partir de trocas de calor proveniente da diferença de temperatura entre dois meios, gerando um fluxo de calor. Dessa forma a saída do sistema, tida como T_o é o resultado da integração da soma de todas as parcelas de fluxo de calor, representadas pelo fluxo do ar condicionado, da sala vizinha, da sala externa e da radiação.

O processo é não-linear devido à chave que ora alimenta o sistema com a temperatura da serpentina, ora com a temperatura da sala; dependendo da referência.

Analisando agora a malha que representa a umidade relativa, sabe-se que esta está diretamente relacionada com a temperatura. Logo se torna complexo lidar diretamente com a umidade relativa pois uma alteração na temperatura resulta em uma variação da umidade, sem que haja alteração na quantidade de água da sala. Para contornar este problema, se utiliza como variável a umidade absoluta, que é ortogonal à temperatura. ? ortogonal?? Referência?

Utiliza-se das Equações em 4.1 para modelar o bloco que faz a conversão de temperatura e umidade relativa para umidade absoluta sem a necessidade do uso da carta psicrométrica. Um modelo semelhante porém com lógica inversa faz a conversão de umidade absoluta para relativa.

$$AH = \frac{e}{R_v T} \quad (4.1)$$

tendo e e e_w como

$$e = e_w \cdot \left(\frac{\phi_{ur}}{100} \right) \cdot f \quad e_w = A \cdot e^{\left(\frac{mT}{T+T_n} \right)}$$

Todas as variáveis devem ser descritas. T, phi_ur, m ??

assim, AH é a umidade absoluta, e e e_w é a pressão do vapor d'água e pressão de saturação do vapor d'água respectivamente, A , T_n e R_v são constantes e f é uma função da pressão atmosférica, que no caso de Brasília é contante 1017 hPa.

A Figura 4.3 mostra o sistema com foco na parte evaporativa do processo, com os blocos de conversão de umidade.

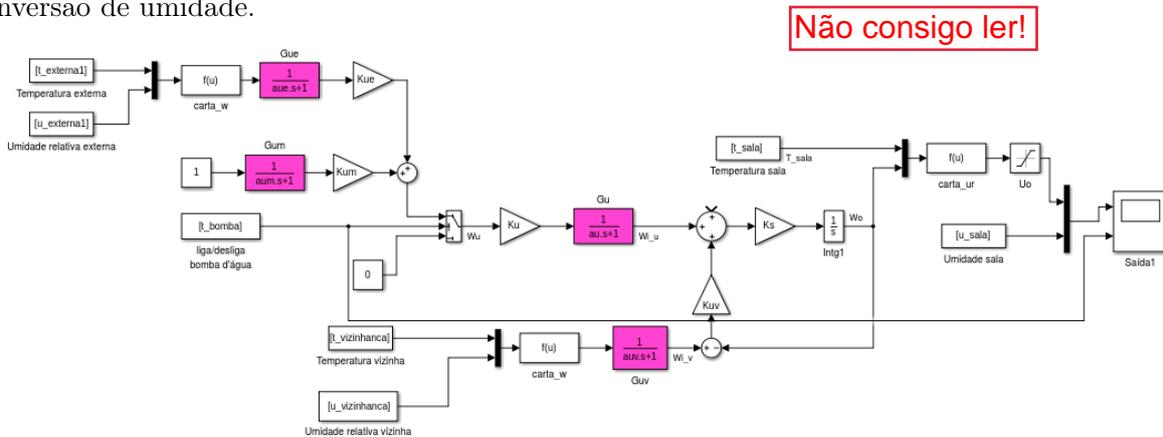


Figura 4.3: Modelo evaporativo

O modelo completo do sistema leva em conta também a relação de uma malha sobre a outra que no caso foi representado pelos ganhos K_{tu} e K_{ut} . A Figura 4.4 mostra o modelo completo com as duas malhas.

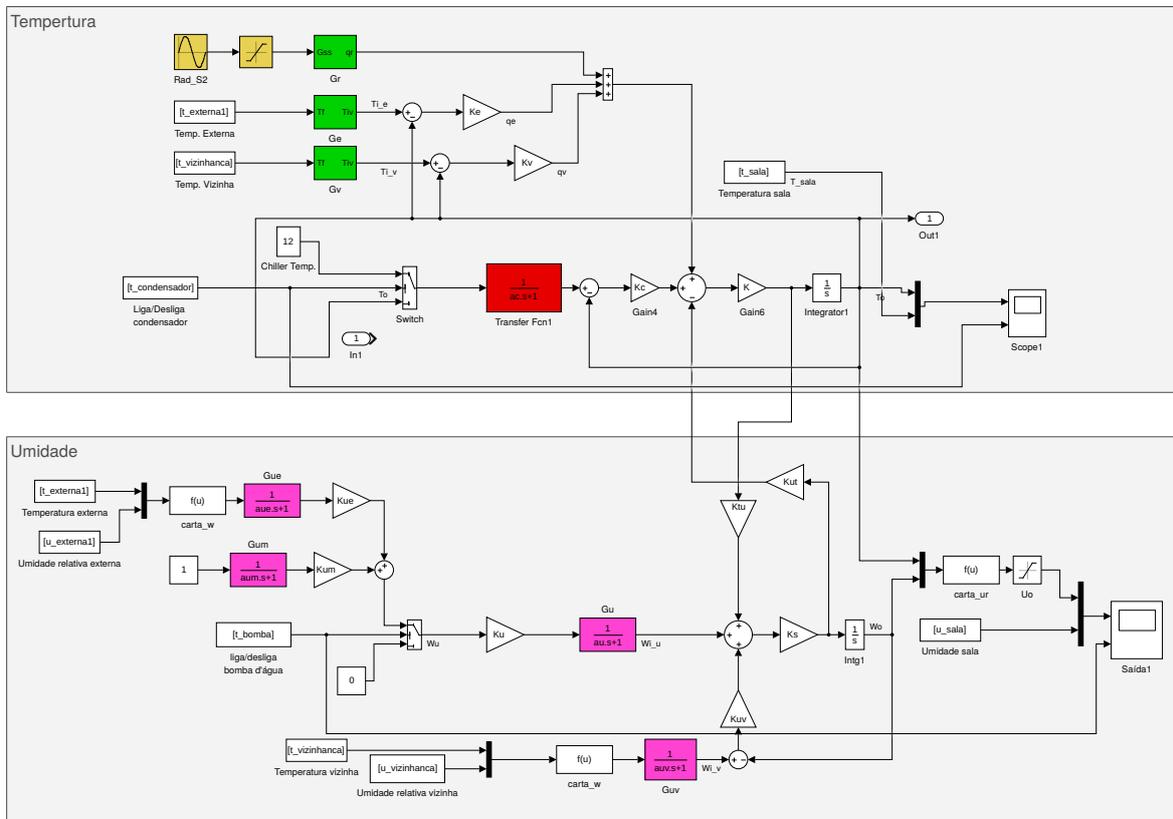


Figura 4.4: Modelo completo

4.1.1 Simulação modelo

A modelagem feita foi do tipo caixa cinza que utiliza, como citado anteriormente, dos princípios fundamentais que se baseiam no fluxo de calor e condução mássica da água. Todas as constantes de tempo e os ganhos dos processo foram utilizados conforme os calculados em [5], porém com algumas alterações tanto na estrutura quanto nos valores, visto que o modelo não estava tão preciso.

A Figura 4.5 mostra a simulação do modelo completo com a comparação do modelo com a medição real para um acionamento do tipo Liga-Desliga do compressor de forma aleatória durante um pouco mais de um dia, entre 19 e 21 de Outubro.

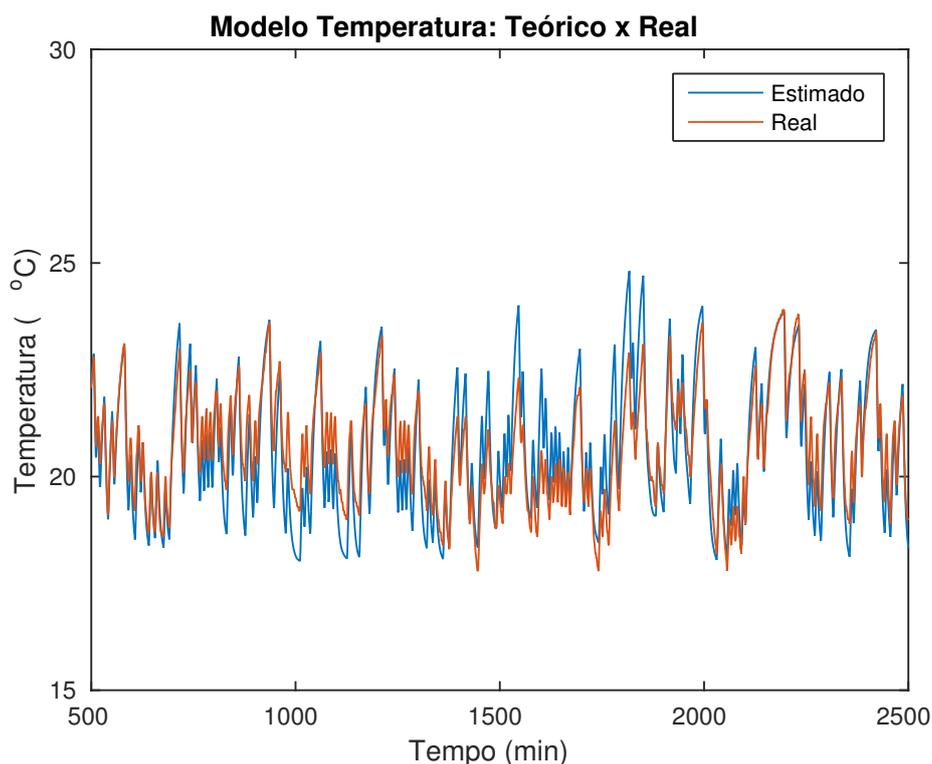


Figura 4.5: Resultado da simulação para temperatura

Pode-se dizer que o modelo para a temperatura está satisfatório visto que o modelo teórico acompanha a dinâmica real medida da sala. O modelo da umidade, porém, tem algumas imperfeições que eram esperadas pois possui uma dinâmica mais complexa que o modelo de temperatura. Pode-se ver que as curvas tem uma variação próximas uma das outras e em alguns momentos des-toam de maneira mais agressiva; por isso o controlador deverá ser robusto e independente das variações ou imperfeições do modelo.

A Figura 4.6 mostra o resultado da simulação para o mesmo tempo mostrado na Figura 4.5 porém com relação à umidade. É importante ressaltar também que o fluxo de pessoas e consequente abertura da porta afetam os dados de maneira não esperada, que não podem ser modelados e se refletem nas curvas.

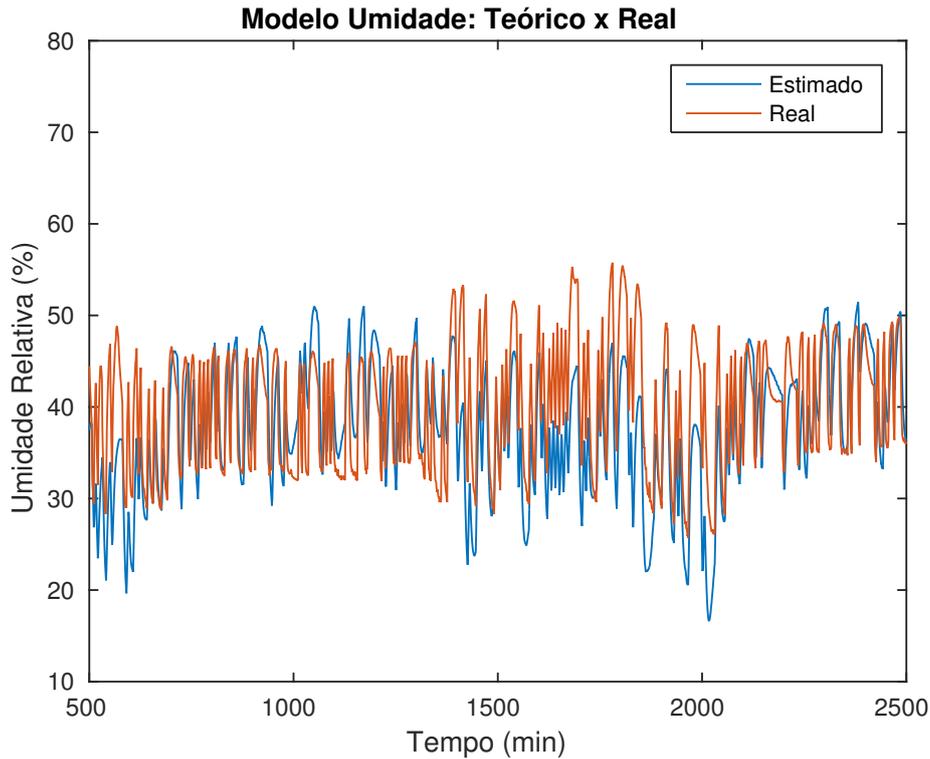


Figura 4.6: Resultado da simulação para umidade

4.2 Sistema TITO

Como já visto, o objeto de estudo deste trabalho se encontra no controle multivariável. Anteriormente, foram feitos projetos de controle com foco apenas na temperatura, ou seja, *single-input-single-output* (SISO). Para este caso, entretanto, temos duas variáveis que devem ser controladas com duas malhas, configurando assim um sistema do tipo *multiple-input-multiple-output* (MIMO), mais precisamente um *two-input-two-output* (TITO), de maneira que uma entrada não afeta uma só saída e uma saída pode afetar outras entradas.

A Figura 4.7 ilustra a interação entre um sistema TITO em forma de diagrama de blocos semelhante ao sistema de climatização híbrida.

A saída y_1 se relaciona com as entradas u_1 e u_2 por meio das funções de transferência G_{11} e G_{21} , respectivamente, enquanto a saída y_2 se relaciona com as entradas u_1 e u_2 por meio das funções de transferência G_{12} e G_{22} , respectivamente.

O sistema MIMO da Figura 4.7 também pode ser representado matematicamente na forma matricial. A expressão 4.2 traduz o sistema 2x2 estudado.

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (4.2)$$

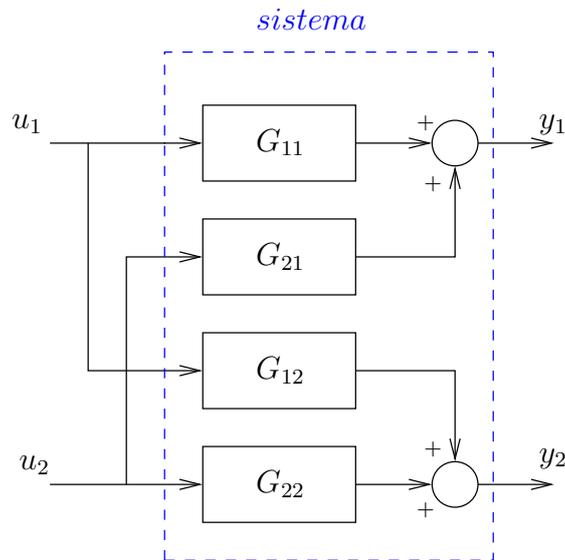


Figura 4.7: Estrutura TITO

4.3 Matriz de ganhos relativos

Em sistemas do tipo MIMO como o de climatização híbrida se faz necessário o estudo das interações de entradas das malhas nas respectivas saídas assim como nas malhas adjacentes. Sabe-se que, a partir do modelo do sistema, deve-se analisar as variáveis manipuladas e controladas; neste caso as variáveis controladas são umidade e temperatura; já as manipuladas são o compressor, a bomba d'água e a abertura do *dampner*. O ventilador é mantido sempre ligado pois é comum aos dois modos de operação e garante a circulação contínua do ar.

Para sistemas onde não se sabe exatamente qual das variáveis de entrada vai ter uma influência maior em uma determinada variável de saída, calcula-se a matriz de ganhos relativos, RGA (*Relative Gain Array*). A RGA é uma matriz de ganhos adimensionais que ilustra o grau de acoplamento de uma dada saída em relação a cada entrada. Essa relação é importante pois indica quais são os pares de variáveis de controle que devem ser escolhidos.

Os coeficientes da matriz podem ser obtidos tanto experimentalmente quanto pelo modelo de estado estacionário do sistema se este for conhecido. Desta forma, os ganhos relativos são a razão entre o ganho estacionário de cada saída por cada entrada se as demais entradas do sistemas são mantidas constantes e o ganho estacionário de cada saída por cada entrada se as demais saídas em questão forem mantidas constantes [16]. Por exemplo, o ganho λ_{11} da matriz pode ser obtido conforme a Equação 4.3.

$$\lambda_{ij} = \frac{k_{ij}|_{u_2}}{k_{ij}|_{y_1}} = \frac{(\Delta y_i / \Delta u_j)|_{u_2}}{(\Delta y_i / \Delta u_j)|_{y_2}} \quad (4.3)$$

Não seriam 11 em vez de i,j?
Parece que falta algo!
i=1,2; j=1,2.

onde o ganho λ_{11} é obtido pela razão entre a variação Δy_1 dado um degrau na entrada u_1 de amplitude Δu_1 para a entrada u_2 mantida contante; da mesma forma a variação Δy_1 deve ser obtida aplicando-se um degrau na entrada u_1 de amplitude Δu_1 para a saída y_2 mantida contante.

Partindo do modelo, é possível obter a RGA de maneira analítica pelos ganhos em estado estacionário, diferente do método experimental descrito acima. Então tendo o sistema expresso em 4.4.

$$y_1 = K_{11}u_1 + K_{12}u_2 \quad (4.4)$$

$$y_2 = K_{21}u_1 + K_{22}u_2 \quad (4.5)$$

Deseja-se achar os ganhos λ_{ij} de acordo com as variações da saída em relação à entrada, desde que as outras entradas e saídas sejam constantes. Dessa maneira, o primeiro ganho é dado por:

$$k_{11}|_{u_2} = \frac{\delta y_1}{\delta x_1}|_{u_2} = K_{11} \quad (4.6)$$

o segundo ganho então pode ser obtido substituindo u_2 da Equação 4.5 na Equação 4.4, tendo:

$$y_1 = K_{11}u_1 + \frac{K_{21}(y_2 - K_{21}u_1)}{K_{22}} \quad (4.7)$$

$$k_{11}|_{y_2} = \frac{\delta y_1}{\delta x_1}|_{y_2} = K_{11} - \frac{K_{21}K_{21}}{K_{22}}$$

portanto, o ganho relativo é:

$$\lambda_{11} = \frac{1}{1 - (K_{21}K_{21})/(K_{11}K_{22})} \quad (4.8)$$

Segue de maneira análoga o cálculo de λ_{12} , λ_{21} e λ_{22} . A partir de [17], tem-se que para um sistema 2x2, dado que a matriz K de ganhos do processo é conhecida, a RGA do sistema pode ser calculada da seguinte maneira:

$$\Lambda = \begin{bmatrix} \lambda & 1 - \lambda \\ 1 - \lambda & \lambda \end{bmatrix} \quad (4.9)$$

onde

$$\lambda = \frac{G_{11}(0)G_{22}(0)}{G_{11}(0)G_{22}(0) - G_{12}(0)G_{21}(0)}$$

em que $G_{ij}(0)$ é o elemento ij da matriz de transferência do sistema. A soma de todos elementos de qualquer matriz ou de qualquer coluna deve sempre resultar em 1, como visto na Equação 4.9.

Analisando os possíveis valores de λ , pode-se escolher o melhor emparelhamento de acordo com os resultados. Valores de λ_{ij} próximos de zero mostram que o par escolhido tem pouca interação ou nenhuma. Por outro lado, valores próximos a 1 mostram que o par em questão tem uma boa interação e não tem forte influência de outras malhas.

Se dois ganhos relativos na mesma linha ou coluna são iguais a 0,5, então o efeito das duas entradas em uma saída ou de uma entrada em duas saídas é igual. Nesse caso, não importa a escolha de emparelhamento de variáveis, pois o grau de interação é o mesmo.

Simplificando, a regra de emparelhamento que deve ser usada é que as malhas de controle devem usar pares de entrada e saída cujo ganho relativo seja positivo e o mais próximo da unidade possível. A Tabela 4.1 mostra um resumo do significado da matriz.

Tabela 4.1: Valores de λ_{ij} [7]

Valores de λ_{ij}	Significado
$\lambda_{ij} > 1$	Há interação entre as malhas de controle
$\lambda_{ij} < 0$	Impossível de controlar o sistema. Deve ser evitado no emparelhamento
$\lambda_{ij} = 1$	É um par ideal, ou seja somente a entrada i atua em j
$\lambda_{ij} = 0$	A variável de entrada i não atua sobre j
$0 < \lambda_{ij} < 1$	Há interação entre as malhas kB

Para o sistema híbrido tem-se a seguinte matriz de função de transferência, onde para ilustração se adotou uma realização mínima:

$$G(s) = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} = \begin{bmatrix} \frac{0.1293}{s^2 + 1.794s + 0.264} & \frac{0.0001509s}{s^3 + 1.84s^2 + 0.3465s + 0.01215} \\ \frac{-1.516e^{-05}s^2 - 2.452e^{-06}s}{s^4 + 2.196s^3 + 0.8723s^2 + 0.08355s + 0.002207} & \frac{0.02276s + 0.03905}{s^3 + 2.117s^2 + 0.7061s + 0.02811} \end{bmatrix} \quad (4.10)$$

logo

$$G(0) = \begin{bmatrix} 0.4897 & 7.4025e^{-18} \\ 2.2063e^{-18} & 1.3892 \end{bmatrix}$$

dos ganhos estacionários das funções de transferência, temos a seguinte matriz RGA, para uma realização não mínima:

$$\Lambda = \begin{bmatrix} 1.0 & 2.2849e^{-34} \\ -4.0011e^{-36} & 1.0 \end{bmatrix} \quad (4.11)$$

onde o melhor emparelhamento pode ser ilustrado pela seguinte matriz:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (4.12)$$

Ou seja, de acordo com a RGA o melhor emparelhamento seria, conforme já esperado, $u_1 \rightarrow y_1$ e $u_2 \rightarrow y_2$.

4.4 Desacoplamento

Boa parte dos sistemas multivariáveis que apresentam interação entre suas entradas e saídas não podem ser controlados utilizando estratégias mais simples do tipo SISO, como a descentralizada por exemplo. O controle eficiente desses processos pode se tornar um problema complexo onde não cabem técnicas já consolidadas.

Uma estratégia bastante utilizada para tratar o acoplamento entre malhas de um sistema MIMO é o método do desacoplamento; este tem como objetivo minimizar o efeito de uma variável sobre a outra podendo assim fazer um estudo mais simplificado, projetando controladores para certo par entrada-saída.

Baseado nos resultados da RGA apresentados na seção 4.3 pode-se escolher um melhor par para o acoplamento. É possível então implementar um sistema de controle com uma matriz de desacoplamento que combina um compensador $C(s)$ com um compensador $D(s)$ de modo que o controlador visualize o processo aparente $Q(s)$ como um conjunto de processos completamente independentes. Portanto, a essência dessa estratégia é introduzir dinâmicas que cancelem as interações entre as variáveis de processo, permitindo um controle independente [18].

A Figura 4.8 mostra a estrutura convencional em diagrama de blocos de um desacoplador para o sistema de duas entradas e duas saídas semelhante ao sistema híbrido.

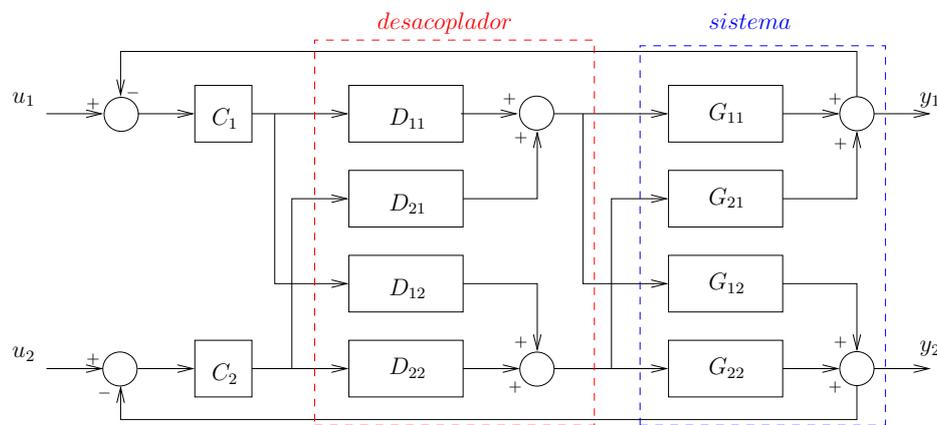


Figura 4.8: Sistema multivariável com desacoplador completo

Onde os blocos compensadores D_{11} , D_{12} , D_{21} e D_{22} são os elementos de $D(s)$; C_1 e C_2 são os controladores correspondentes à $C(s)$ e por fim G_{11} , G_{12} , G_{21} e G_{22} são os elementos da matriz de transferência $G(s)$.

A matriz $D(s)$ deve ser projetada de maneira que o produto de $G(s)$ e $D(s)$ resulte na matriz diagonal $Q(s)$ conforme a Equação 4.13.

$$Q(s) = D(s)G(s) \quad (4.13)$$

onde

$$D(s) = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \quad Q(s) = \begin{bmatrix} Q_{11} & 0 \\ 0 & Q_{12} \end{bmatrix}$$

4.4.1 Desacoplador ideal

Para o caso de um desacoplador ideal como o mostrado na Figura 4.8 proposto por [19] permite que os elementos do controlador C_1 e C_2 sejam sintonizados de acordo com os elementos atribuídos à matriz aparente do processo $Q(s)$. Esta realização, desacoplador ideal, pode se traduzir em uma

maior facilidade no projeto e sintonia do controlador; sua estrutura, no entanto, geralmente leva a expressões complexas de $D(s)$ não realizáveis. A Expressão 4.14 relaciona as matrizes $D(s)$ e $Q(s)$.

$$Q(s) = D(s)G(s)$$

$$D(s) = G(s)^{-1}Q(s) = \begin{bmatrix} G_{22}Q_{11} & -G_{12}Q_{22} \\ -G_{21}Q_{11} & G_{11}Q_{22} \end{bmatrix} \quad (4.14)$$

4.4.2 Desacoplador simplificado

Há também um outro método descrito por [19] como sendo o *Simplified decoupling* (Desacoplador simplificado) que é vastamente utilizado na literatura pois sua maior vantagem em relação ao ideal, também conhecido com completo, é a simplicidade dos elementos envolvidos. A Equação mostra a matriz de desacoplamento simplificada:

$$D(s) = \begin{bmatrix} 1 & -\frac{G_{12}}{G_{11}} \\ -\frac{G_{21}}{G_{22}} & 1 \end{bmatrix} \quad (4.15)$$

que da Equação 4.14 resulta na seguinte expressão para a matriz de transferência $Q(s)$:

$$Q(s) = \begin{bmatrix} G_{11} - \frac{G_{12}G_{21}}{G_{22}} & 0 \\ 0 & G_{22} - \frac{G_{12}G_{21}}{G_{11}} \end{bmatrix} \quad (4.16)$$

No desacoplador simplificado a matriz $D(s)$ possui n elementos iguais a 1, sendo n a dimensão da própria matriz. Como dito anteriormente, o desacoplador simplificado tem como consequência uma maior facilidade no projeto na matriz de desacoplamento $D(s)$, entretanto os elementos de $Q(s)$ podem se tornar complexos visto que seus elementos são somas de funções de transferência; tal característica pode dificultar o projeto do controlador. A Figura 4.9 ilustra o sistema com um desacoplador simples.

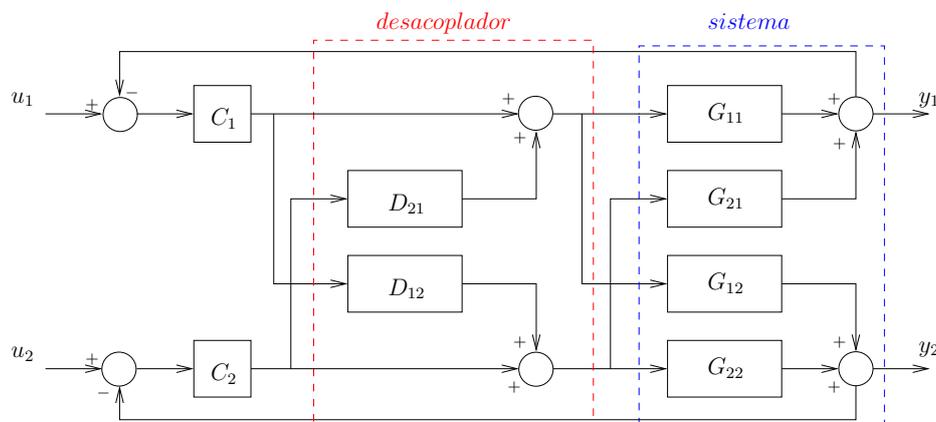


Figura 4.9: Sistema multivariável com desacoplador simples

Aplicando as Equações 4.15 e 4.16 à matriz em 4.10 obtêm-se os seguintes resultados para a matriz de desacoplamento.

$$D(s) = \begin{bmatrix} 1 & \frac{-0.07005s}{s + 27.61} \\ \frac{0.000666s^3 + 0.00125s^2 + 0.0001848s}{s^3 + 3.509s^2 + 3.211s + 0.231} & 1 \end{bmatrix} \quad (4.17)$$

e para a matriz de transferência $Q(s)$.

$$Q(s) = \begin{bmatrix} \frac{0.1293s^2 + 0.4434s + 0.3803}{s^4 + 5.224s^3 + 9.359s^2 + 6.182s + 0.7765} & 0 \\ 0 & \frac{0.02267}{s^2 + 0.4021 + 0.01639} \end{bmatrix} \quad (4.18)$$

onde $Q(s)$ é o sistema enxergado pelas entradas u_1 e u_2 , conseqüentemente pelo controlador também. A Figura 4.10 mostra o modelo no Simulink com os blocos de desacoplamento de $D(s)$ resultantes da Expressão 4.17.

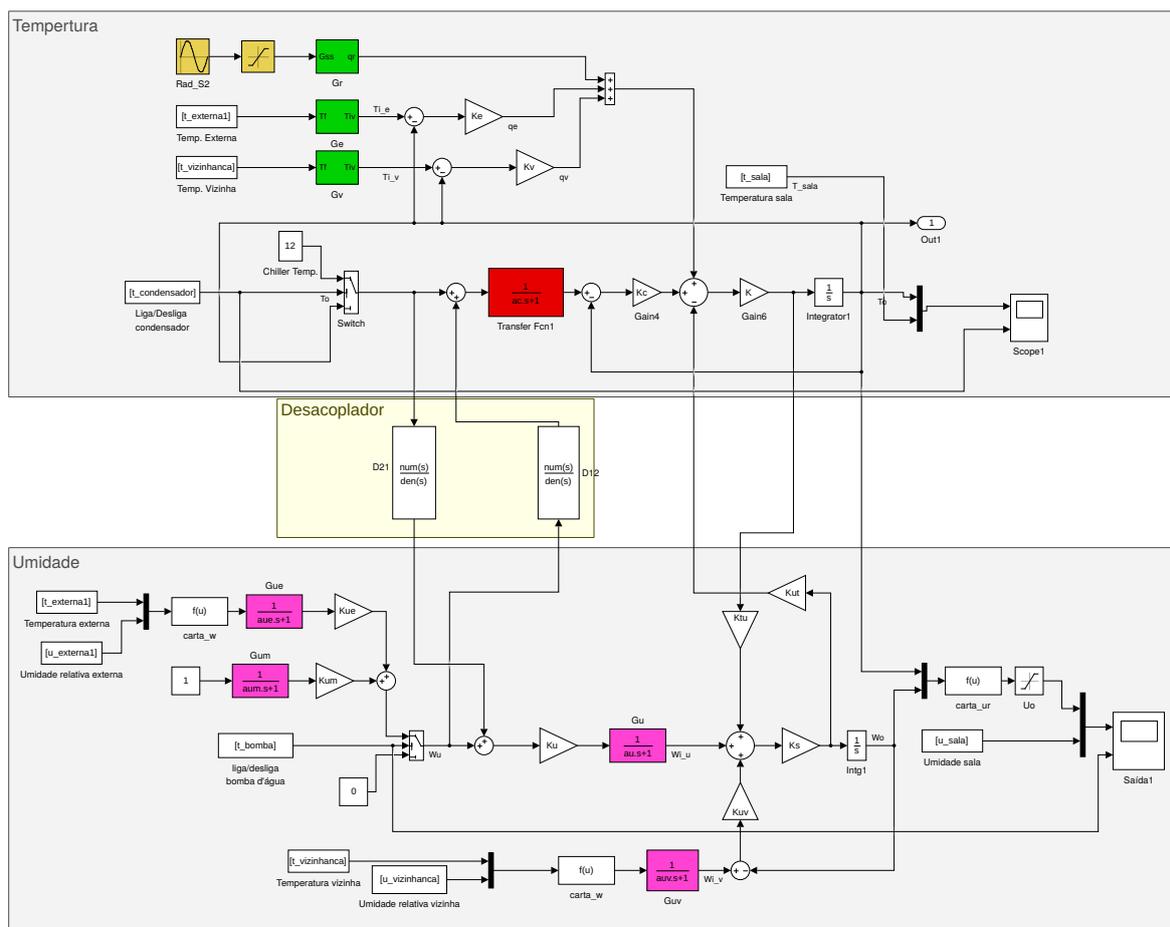


Figura 4.10: Diagrama de blocos no Simulink com desacoplador

Analisando a dinâmica do modelo da Figura 4.10 pode-se obter analiticamente as funções de transferência G_{d11} e G_{d22} que relacionam as entradas u_1 e u_2 às saídas y_1 e y_2 respectivamente. Foi possível observar que G_{d11} e G_{d22} correspondem à Q_{11} e Q_{22} referentes à matriz de transferência $Q(s)$ mostrada pela Equação 4.18.

4.4.3 Discretização desacoplador

A discretização da dinâmica do desacoplador também foi necessária para sua implementação no Arduino. Dessa forma para o desacoplador D_{12} segue que:

$$D_{12} = \frac{-0.07005s}{s + 27.61} \quad (4.19)$$

que para um tempo de amostragem de $T = 0.5$ s.

$$D_{12}^z = \frac{-0.001167z + 0.001167}{z - 0.9773} \quad (4.20)$$

onde pode-se encontrar a equação de diferenças para tal desacoplador, tornando-se fácil sua implementação; segue que

$$\begin{aligned} \frac{D_{12}^z}{z} &= \frac{-0.001167z + 0.001167}{z^2 - 0.9973z} \\ &= \frac{0.0000271062}{z - 0.9973} - \frac{0.001167}{z} \end{aligned} \quad (4.21)$$

aplicando portanto a transformada z inversa $\mathcal{Z}^{-1}\{D_{12}^z\}$, pode se encontrar a seguinte equação de diferenças:

$$D_{12}[k] = 0.0000271062(0.9973)^k - 0.001167 \cdot \delta[k] \quad (4.22)$$

Analogamente, o mesmo pode ser feito para o desacoplado D_{21} , logo seguindo os mesmo passos temos que:

$$D_{21}[k] = 0.0003(1.69466)^k + 0.0003(1.73581)^k + 0.00003(0.07852)^k \quad (4.23)$$

4.5 Projeto controlador

Tendo como principais variáveis manipuladas o compressor, no caso do modo convencional, e a bomba d'água, no caso do modo evaporativo, que só podem assumir valores 1 e 0 que correspondem à ligado e desligado respectivamente. Essa restrição resulta em uma limitação do controlador e conseqüentemente de seu desempenho.

Em vista disso o projeto será feito de maneira normal porém a saída de controle será convertida em um sinal do tipo PWM, ou modulação por largura de pulso, como visto na Seção 3.5. É importante ressaltar também que embora o modelo esteja representado em tempo contínuo o

controlador deverá ser traduzido de maneira discreta, visto que o objetivo é embarcá-lo no Arduino.

4.5.1 Controlador PI

O controlador do tipo PI é um compensador largamente utilizado em várias abordagens devido à sua eficiência e simplicidade de projeto e implementação. A Equação 4.24 mostra a expressão geral para um controlador do tipo PI.

$$C(s) = K_p + \frac{K_i}{s} = K_p \left(1 + \frac{1}{\tau_i s} \right) = K_p \left(\frac{s + K_i/K_p}{s} \right) \quad (4.24)$$

O compensador do tipo PI se torna uma boa opção visto que o processo possui uma dinâmica lenta e que os principais requisitos de projeto são erro nula em estado estacionário e um sobrepasso de 0%. Na prática sabe-se que esses parâmetro de projeto não podem ser completamente atingidos devido a limitação do acionamento que só aceita estados 'ligado' e 'desligado'. Sabe-se também que para preservar a vida útil do compressor recomenda-se que o mesmo tenha um período mínimo de 3,5 ~ 4 min; ou seja, uma vez ligado o compressor só poderá ser ligado novamente após 4 minutos aproximadamente. Na prática esta limitação restringe o período do PWM a 4 minutos nos mínimo.

O projeto foi feito baseado no modelo utilizando o método do lugar geométrico das raízes sem levar em conta a não linearidade do processo devido a chave, como já mostrado na Figura 4.10. Analisando primeiro a malha referente ao compressor sabe-se que, conforme visto na Seção 4.4.2, a função de transferência enxergada pelo controlador é dada pelo termo q_{11} da matriz $Q(s)$ na Equação 4.18, logo:

$$q_{11} = \frac{0.1293s^2 + 0.4434s + 0.3803}{s^4 + 5.224s^3 + 9.359s^2 + 6.182s + 0.7765} \quad (4.25)$$

com os zeros e polos dados por:

$$z = \begin{bmatrix} -1.7152 \\ -1.7152 \end{bmatrix} \quad p = \begin{bmatrix} -1.7152 \\ -1.7152 \\ -1.6320 \\ -0.1617 \end{bmatrix}$$

Dessa forma pode-se obter o LGR do sistema dinâmico já alocando um pólo na origem do sistema proveniente do controlador PI.

Pode-se então alocar o zero do controlador conforme o necessário, sempre analisando o LGR resultante. Da Figura 4.11 nota-se que rapidamente o sistema pode ficar instável a depender do incremento do ganho do compensador. Para este caso ainda sem zero e somente com um pólo na origem o ganho crítico é de $K = 3.67$.

A estratégia para alocação do zero foi a de cancelar o pólo mais lento, logo foi colocado em $z = -0.1617$. A partir daí o ganho foi sendo alterado gradativamente de modo que atingisse os

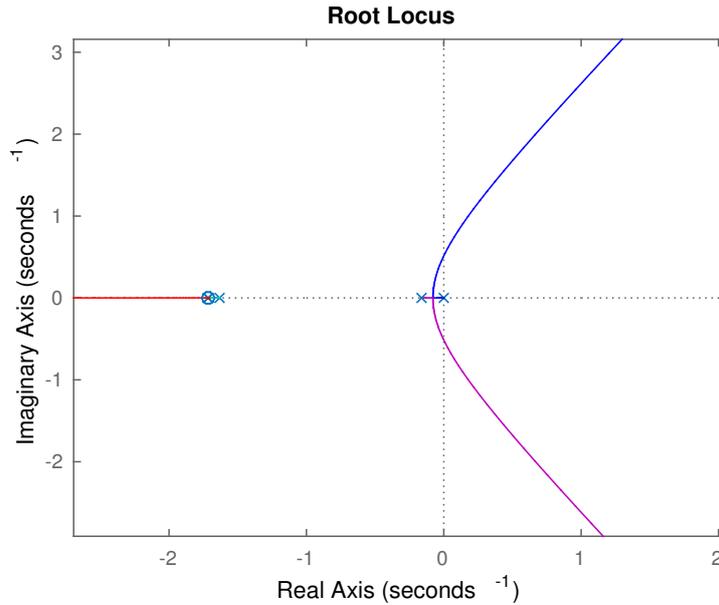


Figura 4.11: LGR da malha de temperatura

requisitos do sistema de 0% de sobrepasso. Foi simulado então o controlador no Simulink e como o esperado foi necessário fazer algumas alterações no ganho quando se adicionou a chave e o módulo PWM. A Equação 4.26 mostra o controlador projetado, assim como os ganhos.

$$C(s) = 1.10178 \left(\frac{1 + 6.2s}{s} \right) \tag{4.26}$$

$$K_p = 1.02 \quad K_i = 0.165$$

O sistema para o controle foi desenhado a princípio para comparar o sinal de controle gerado pelo PI, pelo PWM e por um Liga-Desliga tradicional assim como a potência, conforme a Figura 4.12.

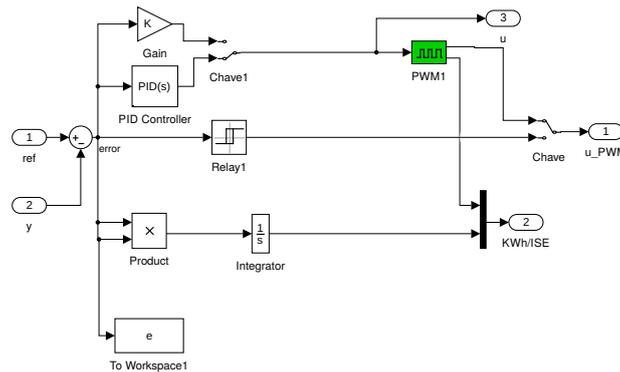


Figura 4.12: Unidade de controle Simulink

O processo foi simulado tendo como referência um *setpoint* operacional de 22.5 que oscila conforme uma onda quadrada. A Figura 4.13 mostra o resultado da simulação, mostrando que o controle está satisfatório; as oscilações em torno do ponto de operação são esperadas para sistemas de refrigeração.

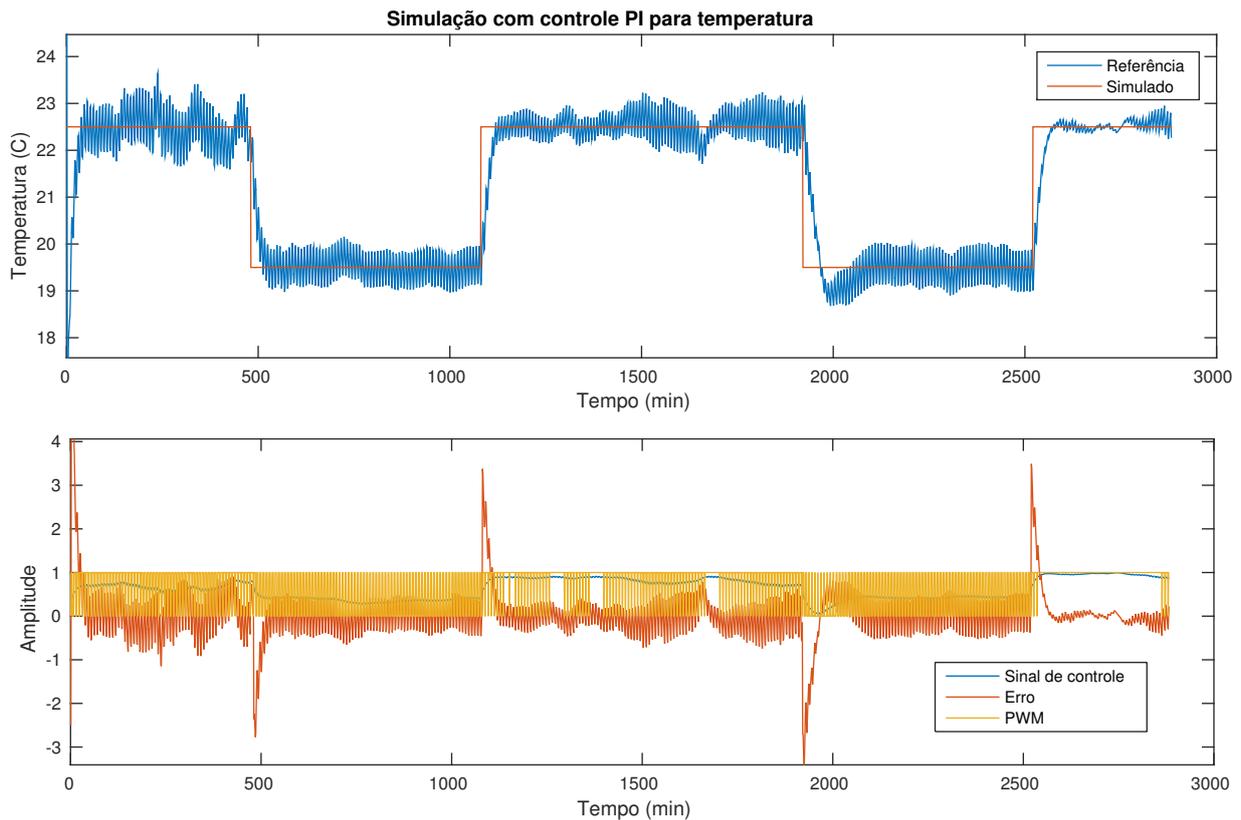


Figura 4.13: Simulação controle PI para temperatura

A discretização do controlador PI para a implementação do código resultou nos mesmos valores de ganhos K_p e K_i do sistema em tempo contínuo para um tempo de amostragem de 0.5 segundo que foi o tempo definido no código.

Em relação ao resfriamento evaporativo, sabe-se que a dinâmica desse processo é bem lenta, por volta de 30 minutos. Dessa maneira, o resfriamento por evaporação será utilizado quando em condições favoráveis, conforme dito no Capítulo 2, em situações quando se pode ter uma real queda da temperatura, geralmente em dias quentes e seco onde a TBU externa está consideravelmente abaixo da TBS interna.

4.5.2 Controle umidade

O controle da umidade se dará durante ciclos a fim de manter uma umidade contante dentro da sala durante o dia, principalmente durante e após longos períodos de uso do sistema de refrigeração convencional que tem como consequência uma queda da umidade relativa. Também será utilizado para a umidificação do ambiente no caso de a umidade abaixar demais, em geral abaixo de 40%, que seria a umidade relativa mínima para uma situação de conforto; o ideal é uma umidade em torno de 50%.

Diferente de trabalhos feitos anteriormente, pode-se ocorrer o funcionamento simultâneo dos dois modos de operação por um breve intervalo de tempo, o que pode levar a uma queda de

desempenho. Da mesma forma, o *damper* poderá ficar aberto durante os dois modos afim de garantir tanto a circulação do ar já refrigerado e umidificado assim como a renovação do ar interno ao se captar o ar externo. Será evitado uso da bomba por mais de 30 minutos consecutivos a fim de minimizar o desperdício de água.

Capítulo 5

Resultados

Este capítulo aborda os resultados obtidos da implementação do controle e da automação do sistema híbrido

5.1 Temperatura

Para os resultados referente ao controle da temperatura foi utilizado um setpoint de 23 °C, que é a temperatura de conforto térmico, com algumas alterações aleatórias nesse setpoint a fim de simular diferentes entradas de um usuário em relação à temperatura desejada.

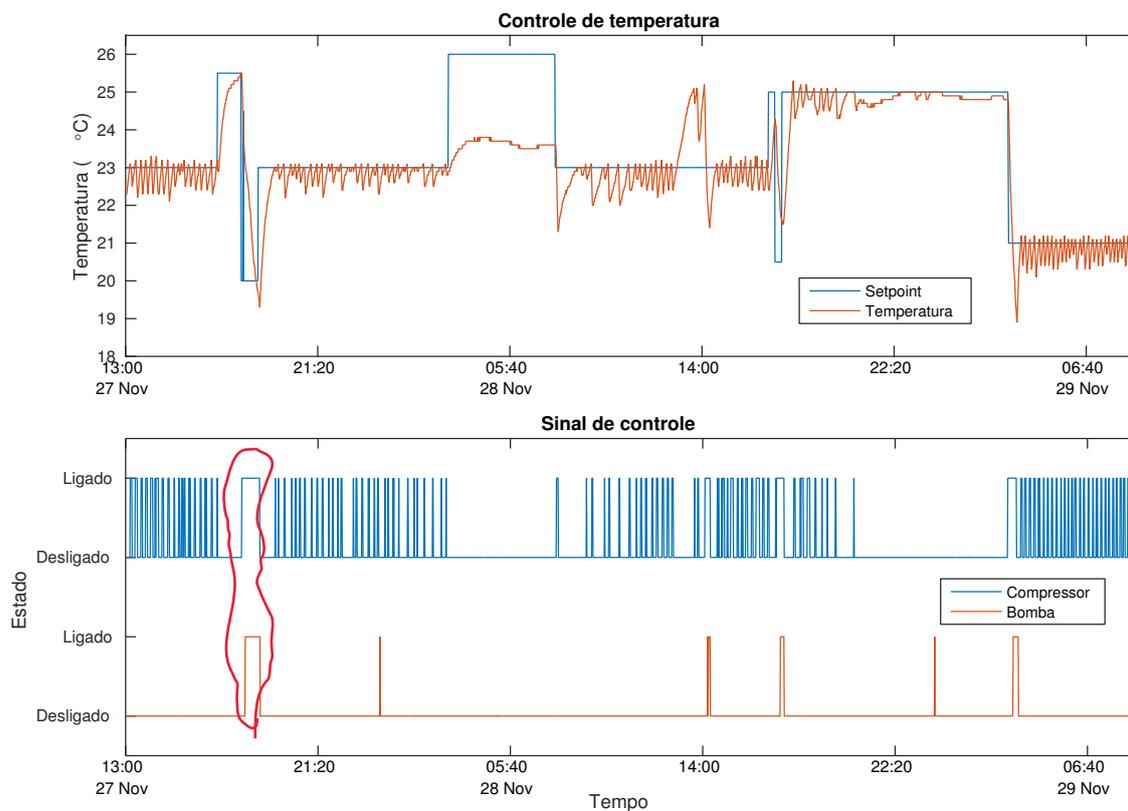


Figura 5.1: Comportamento do controle de temperatura

Tem sentido
Compressor ligado
e bomba d'água?
Damper aberto ->
ar vem de dentro
da sala!

A Figura 5.1 mostra o resultado dos testes para um pouco mais de um dia, iniciado às 13:00:00 do dia 27 de Novembro; sendo o gráfico superior o da temperatura da sala de reuniões e o inferior o sinal dos atuadores. Nota-se que a temperatura segue relativamente bem o *setpoint* da temperatura. Da forma que foi projetado o acionamento por PWM, resultou em uma média de temperatura um pouco abaixo do *setpoint*, o que era esperado visto que o controlador foi implementado com uma resposta mais rápida quando a temperatura interna está acima do *setpoint* e mais lenta no caso contrário. Neste primeiro experimento utilizou-se o controle com ênfase no compressor afim de manter a temperatura bem próxima do *setpoint*, o que justifica a diferença no uso da bomba e do compressor.

Por volta das 05:00 do dia 28 percebe-se que o a temperatura não acompanhou o *setpoint* pois a temperatura externa estava relativamente baixa visto que era o período da noite. Por volta das 14:00 horas do dia 28 hora nota-se também um comportamento estranho para a temperatura. Este comportamento é consequência de sobrecargas na rede elétrica do LARA que levam o Arduino a reiniciar; este está programado para setar o *setpoint* em 25 sempre que é reiniciado, fato que não é notado pelo openHAB até que ocorra uma nova alteração do *setpoint*.

Do sinal de controle nota-se que o PWM foi executado de maneira satisfatória, embora é possível notar pontos onde duas transições seguidas para o estado ligado ocorreram em um intervalo de menos de 3.5 minutos, o que não é recomendável. Este fato ocorre porque durante o ciclo do PWM o sinal de controle pode alterar relativamente bem; dessa forma um ciclo de trabalho pode ter potência de 90% enquanto o próximo de 10%, gerando um chaveamento rápido. O sinal da bomba mostra que ocorreram pequenos ciclos de umidificação da sala.

A Figura 5.2 mostra o comportamento da temperatura interna com um filtro de média móvel em comparação com a temperatura externa.

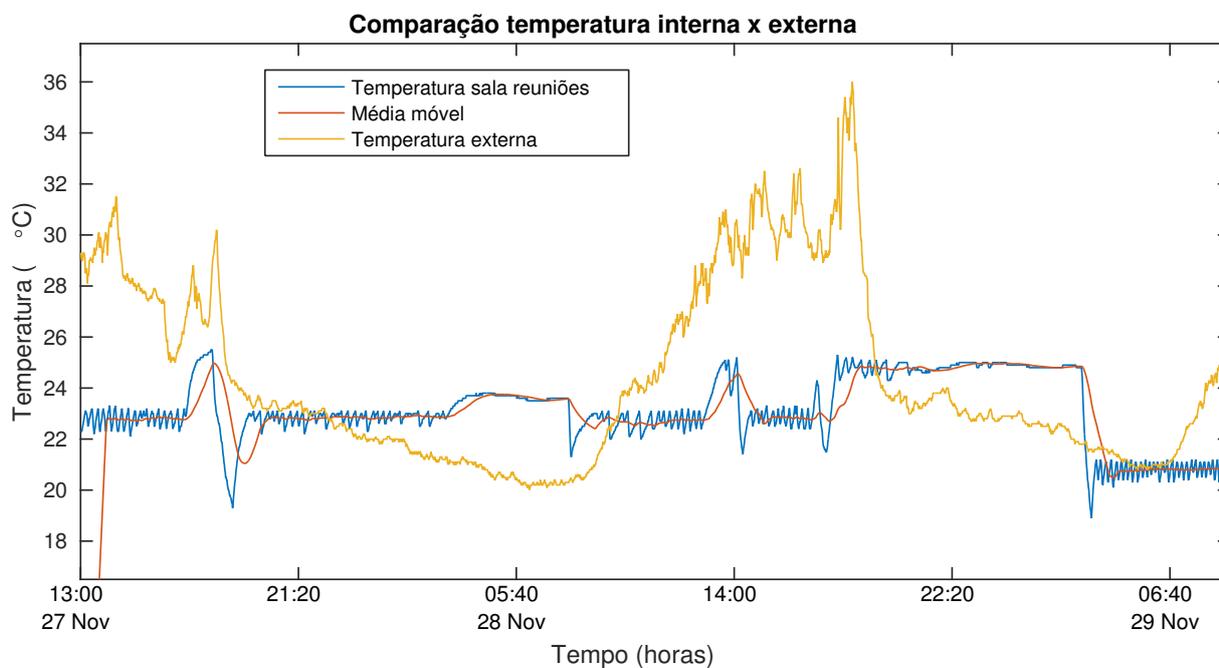


Figura 5.2: Comparação temperatura interna/externa

Nota-se portanto uma amplitude considerável da temperatura externa que não é refletida no ambiente interno da sala de reuniões com exceção dos momentos em que se alterou o *setpoint*. É importante ressaltar que perturbações como a abertura da porta da sala de reuniões podem se refletir nos gráficos como comportamentos inesperados.

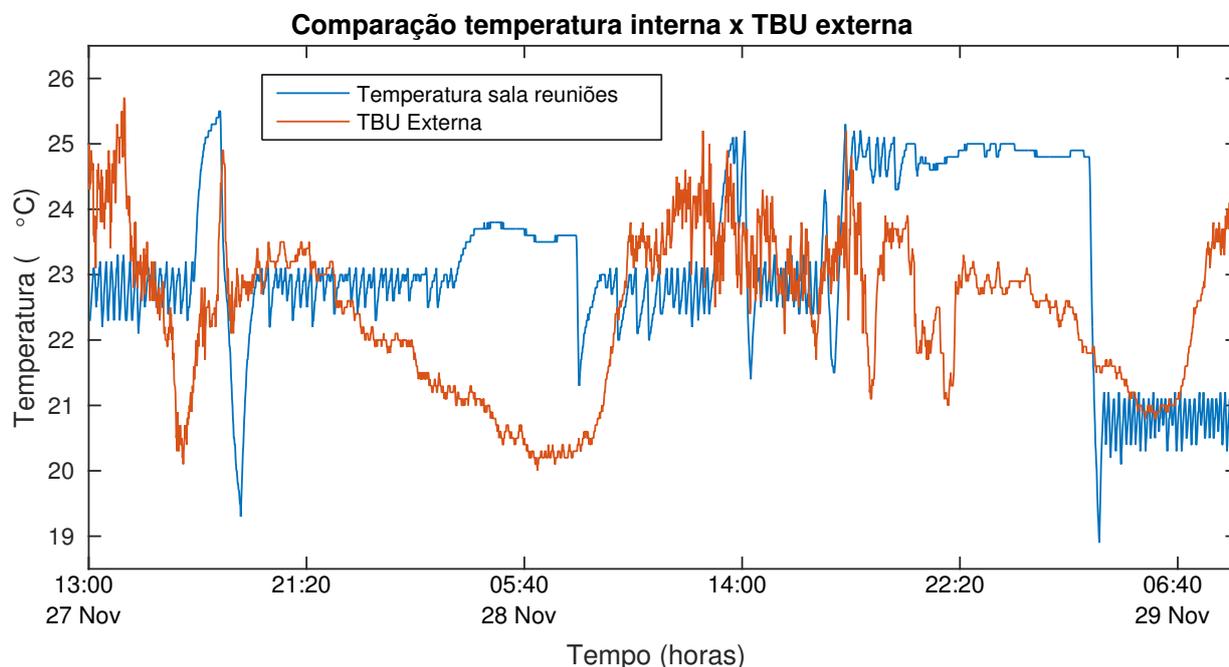


Figura 5.3: Comparação temperatura interna/TBU externa

A Figura 5.3 mostra a relação entre a temperatura interna da sala de reuniões e a TBU externa. Para que o resfriamento no modo evaporativo resulte em um ganho razoável é necessário que a diferença entre a temperatura interna e a TBU externa seja de pelo menos 3°C; ou seja, a TBU externa deve estar pelo menos 3°C acima da temperatura da sala. Em época chuvosa a TBU do ambiente costuma ser elevada e, conforme mostrado pelo gráfico, poucas horas do dia se torna viável a utilização do modo evaporativo.

Para efeito ilustrativo e comparativo, a Tabela 5.1 adaptada de [20] mostra a redução de temperatura teórica para um sistema evaporativo bem instalado:

Tabela 5.1: Redução de temperatura por resfriamento evaporativo

Temperatura externa °C	U.R Externa %	Redução Temp. °C
22 - 25	20% - 45%	10 - 6.5
	50% - 75%	5.5 - 2.5
26 - 30	20% - 45%	11.5 - 6.5
	50% - 75%	6.5 - 3
31 - 33	20% - 45%	12.5 - 8
	50% - 75%	7 - 3

Temperatura externa °C	U.R Externa %	Redução Temp. °C
34 - 37	20% - 45%	13.5 - 8.5
	50% - 75%	6.5 - 3.5
38 - 41	20% - 45%	10 - 6.5
	50% - 75%	7.5 - 3.5
42 - 45	20% - 45%	15 - 9.5
	50% - 75%	-

Em comparação com o sistema instalado na sala de reuniões do LARA porém mostra que o sistema não está funcionando como o esperado. Um problema notado foi a falta de manutenção do sistema o que pode ter reflexo no seu desempenho. A Figura 5.4 mostra um período de 24 horas onde a bomba foi utilizada de forma periódica entre os dias 07 e 08 de Outubro a fim de ilustrar o efeito da refrigeração evaporativa.

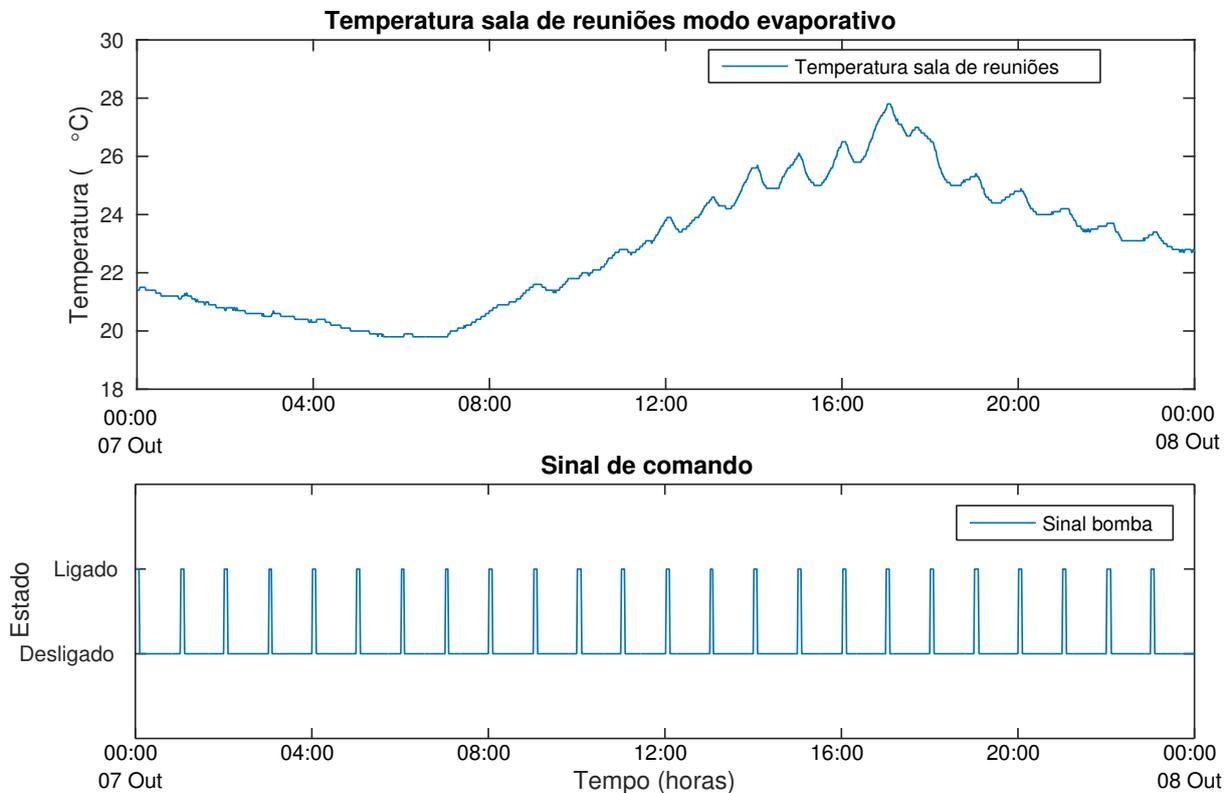


Figura 5.4: Efeito do modo evaporativo na temperatura da sala de reuniões

Nota-se que a temperatura durante o dia tem queda máxima inferior a 4 °C e que a maior variação decrescente de temperatura ocorre durante a noite. Este fato é incompatível com os dados mostrados pela Tabela 5.1, mesmo com as condições externas de alta umidade decorrente do período de chuvas iniciado em Outubro. A Figura 5.5 mostra a relação da temperatura da sala de reuniões e a umidade externa para o mesmo período.

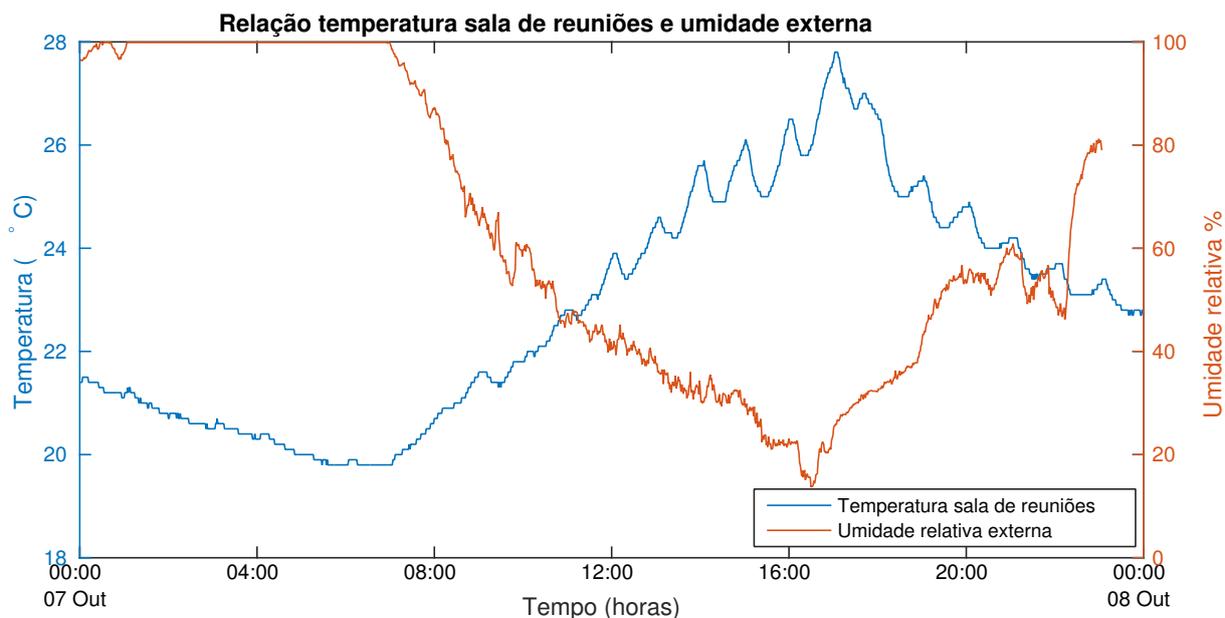


Figura 5.5: Variação temperatura da sala de reuniões e umidade externa

Correlacionando as Figuras 5.4 e 5.5 nota-se que o sistema evaporativo só foi utilizado de maneira mais eficiente por volta das 16:00, quando a umidade relativa do ar externo foi inferior a 20% e com a temperatura interna da sala de reuniões com uma redução de aproximadamente 4°C neste momento; resultado mais satisfatório porém ainda incompatível com os dados da Tabela 5.1.

De posse destes resultados se propôs um controle variável que pudesse ser mudado de acordo com as condições do ano. Essas adaptações podem ser implementadas tanto pelo usuário por meio do supervisor openHAB quanto diretamente no algoritmo do Arduino.

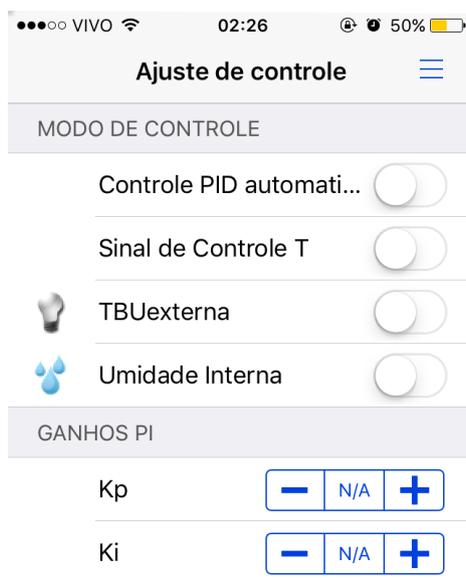


Figura 5.6: Interface de controle

A Figura 5.6 mostra como ficaria a interface por exemplo para mudar tanto o tipo de controle assim como os ganhos do controlador PI *on the fly* sem a necessidade de reprogramar o microcontrolador assim como alterar o approach do sistema.

Levando em conta então as dificuldades do sistema evaporativo como opção principal de resfriamento, optou-se por uma nova estratégia onde se diminuiu o ganho K_p para se ter uma resposta mais lenta do sistema; após a temperatura atingir o *setpoint* utilizou-se o modo evaporativo para se manter à dada temperatura, evitando assim o uso excessivo do compressor.

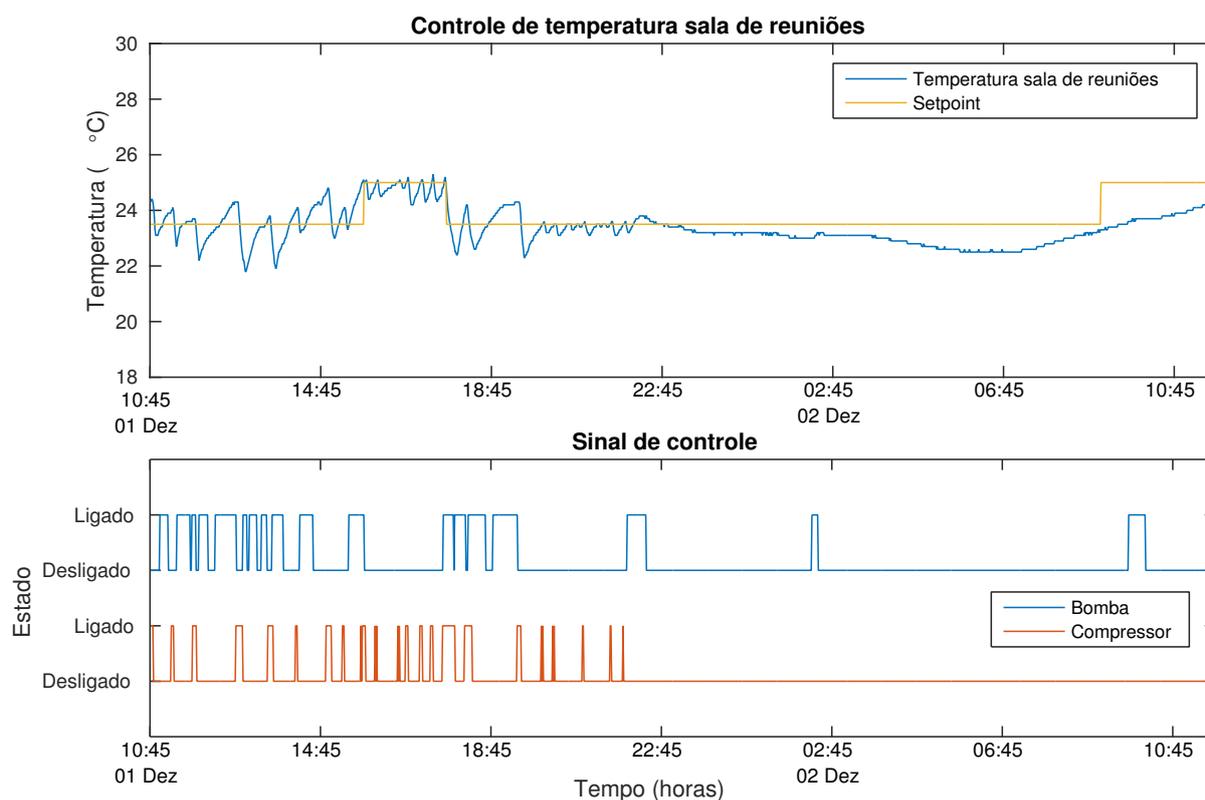


Figura 5.7: Estratégia 2 para controle de temperatura da sala de reuniões

A Figura 5.7 mostra o resultado do experimento rodado na sala de reuniões durante o período de 25 horas entre os dias 1 e 2 de Dezembro. O *setpoint* foi fixado em 23.5 durante a maior parte do tempo com uma variação para 25 em algumas horas do dia. Nota-se que comparando este gráfico com o resultado mostrado na Figura 5.1 tem-se uma maior dificuldade em seguir o *setpoint* com uma oscilação maior em torno desde; algo por volta de 1.5 °C.

Esta estratégia se mostrou pouco eficiente no que concerne a robustez no controle da temperatura da sala de reuniões visto que oscilações acima de 0.5 °C para cima ou para baixo em torno do *setpoint* fogem dos parâmetros de conforto térmico estipulados pela ASHARE. A Figura 5.8 mostra o comportamento da temperatura da sala de reuniões e sua média móvel. Nota-se que na média a temperatura se mostrou perto do *setpoint* como o esperado

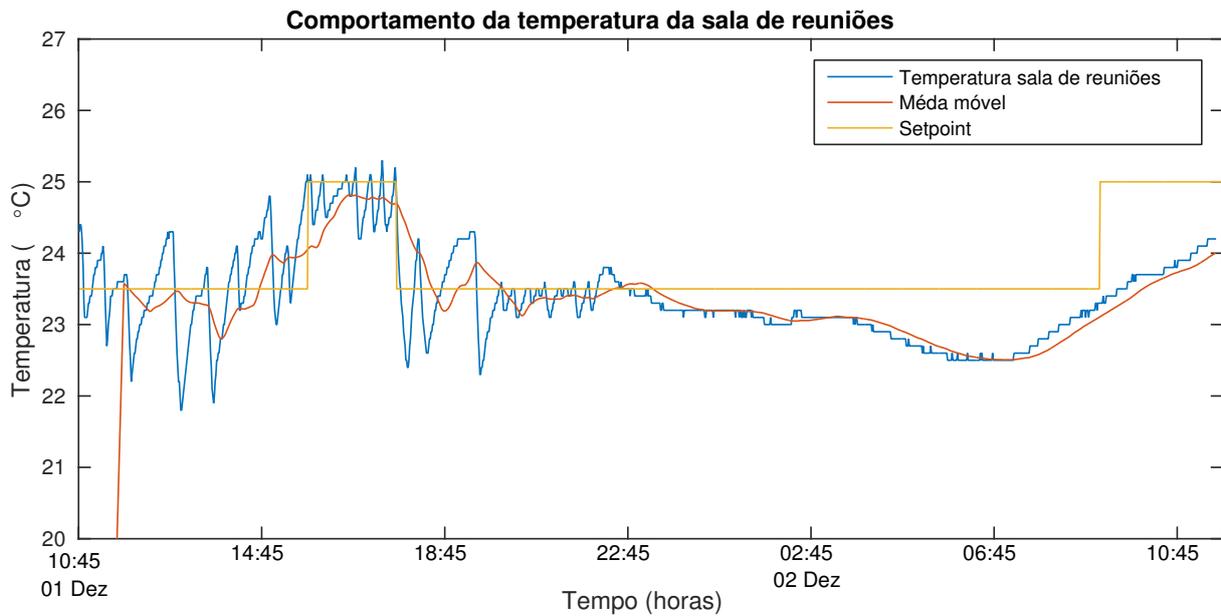


Figura 5.8: Média móvel da temperatura interna da sala de reuniões

Esta estratégia se mostrou pouco eficiente em relação a robustez no controle da temperatura da sala de reuniões; entretanto levando em conta o consumo de energia elétrica esta estratégia se mostrou mais vantajosa se compararmos os gráficos de acionamento do compressor e da bomba d'água apresentados nas Figuras 5.1 e 5.8. Nota-se ao observar os gráficos que o uso do modo evaporativo aumentou consideravelmente ao ponto que o modo convencional decaiu.

Para períodos de aproximadamente 25 horas entre os dias 28 e 29 de Novembro, no caso para a primeira estratégia (Figura 5.1), e 25 horas entre os dias 1 e 2 de Dezembro, para a segunda estratégia Figura 5.8, pode-se relacionar o emprego do modo convencional e evaporativo. A Tabela 5.2 mostra o uso da bomba d'água e do compressor.

Tabela 5.2: Utilização do compressor e bomba d'água

	Compressor		Bomba d'água	
	Ligado	Desligado	Ligado	Desligado
28 - 29	328 minutos	1172 minutos	61 minutos	1439 minutos
	21.8%	78.2%	4%	96%
01 - 02	112 minutos	1388 minutos	313 minutos	1187 minutos
	7.5%	92.5%	20.9%	79.1%

Como já observado anteriormente, a utilização do compressor teve uma diminuição significativa de aproximadamente 65.8%, enquanto que o uso da bomba teve um aumento de aproximadamente 413.1%. Embora o consumo tenha caído, esta não é a opção ótima e seria necessário mais testes para se aprimorar o controle. A Tabela 5.3, obtida por [8], fornece o consumos dos equipamentos nominalmente.

Tabela 5.3: Consumo dos equipamento [8]

Equipamento	Carga [W]	Consumo [kW·h/min]
Unidade condensadora	2180	0.03633
Ventilador	127	0.00212
Bomba d'água	8	0.000133
Sistema de automação	10	0.00016

A partir dos dados das Tabelas 5.1 e 5.3 é possível calcular o consumo para os dois experimentos realizados. O ventilador permaneceu ligado durante a 25 horas a fim de garantir a constante circulação e renovação do ar; assim como o sistema de automação como um todo. Para o período entre 28 e 29 de Novembro obteve-se um consumo total de aproximadamente 15.54 kWh. Já para o período entre 1 e 2 de Dezembro teve um consumo total de aproximadamente 7.53 kWh; ou seja, uma redução de aproximadamente 51.54%. O consumo real aferido pelo medidor de energia Landis + Gyr situado na sala de reuniões mostrou um consumo de 5.45 kWh para o segundo período de testes; esta medição não leva em conta porém o consumo do sistema de automação, no caso o Raspberry Pi, Arduino e roteador. Este consumo é inferior ao consumo de um sistema puramente convencional porém é superior aos resultados obtidos por [6] e [8]; isso se deve principalmente às condições climáticas no período de realização dos experimentos principalmente. Um ponto importante é que não se leva em conta o desperdício de água, visto que esta não é recaptada e sim dispensada para o lado de fora.

5.2 Umidade

Como já dito o controle da umidade foi feito a partir de alguns ciclos de umidificação e principalmente quando a umidade relativa tinha uma diminuição considerável, tipicamente após longo uso do compressor.

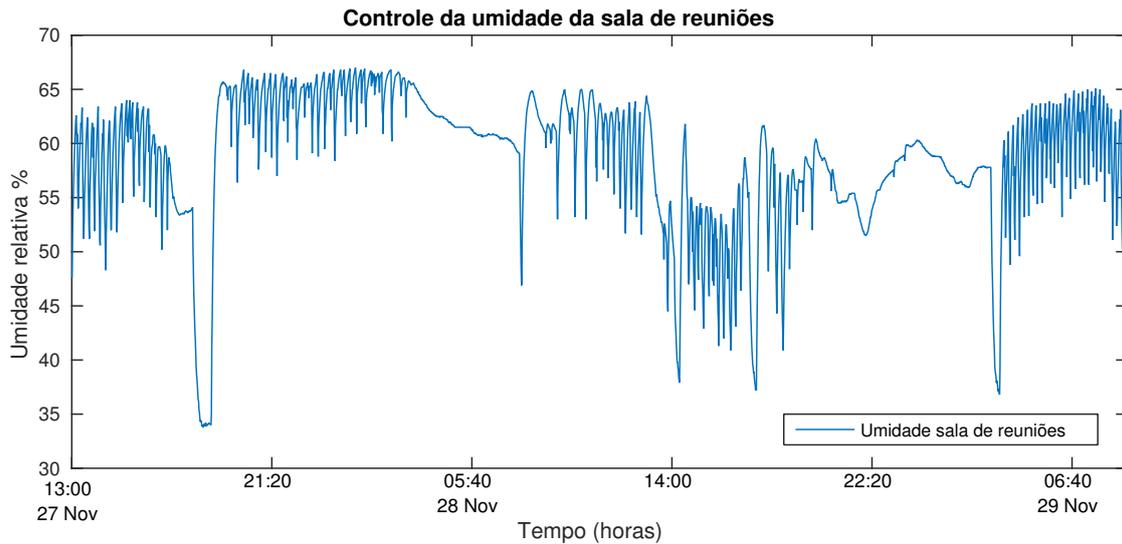


Figura 5.9: Comportamento da umidade interna da sala de reuniões

A Figura 5.9 mostra o resultado da umidade relativa da sala de reuniões para o primeiro experimento, do mesmo modo que foi feito para a temperatura.

Nota-se do gráfico que a umidade relativa interna da sala de reuniões se manteve majoritariamente acima dos 40% e abaixo dos 65% que denomina a faixa para o conforto térmico conforme citado na Tabela 2.3. Perce-se do gráfico da umidade relativa que esta está fortemente relacionada com a temperatura interna, ambas oscilando conforme o mesmo padrão mesmo que a quantidade de água na sala não varie da mesma forma. A Figura 5.10 correlaciona a umidade da sala de reuniões, a externa e da sala vizinha, no caso o próprio Lara.

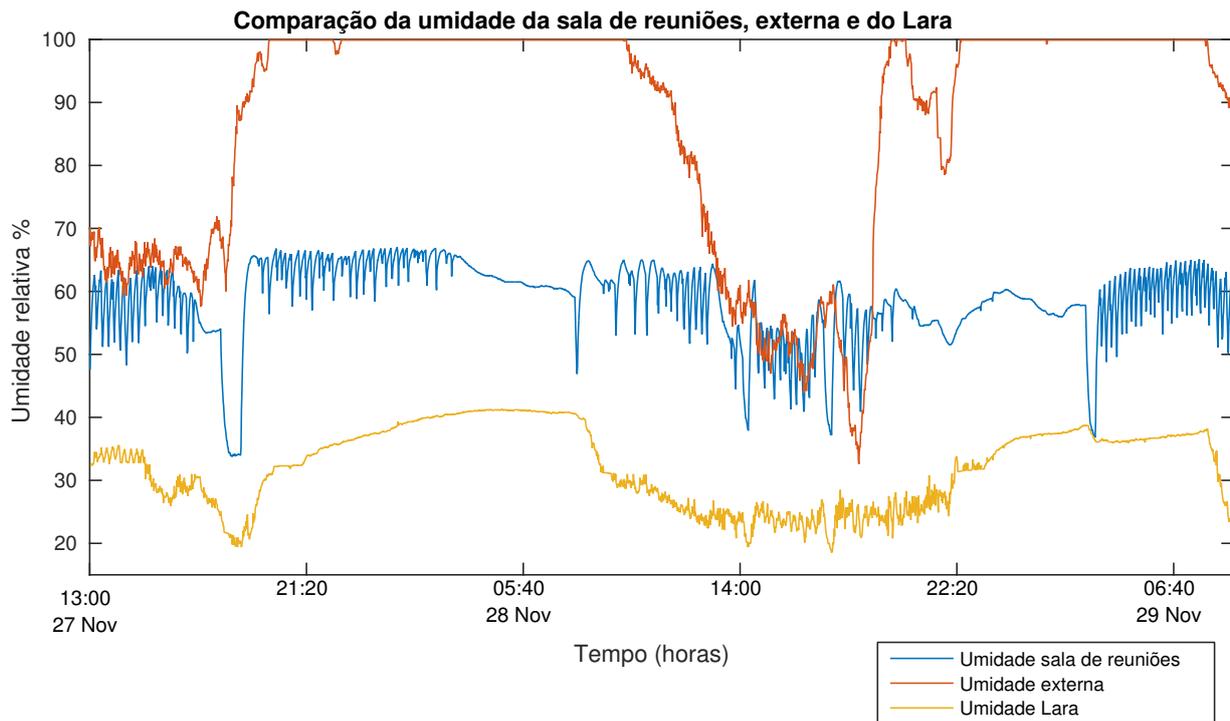


Figura 5.10: Comparação da umidade interna da sala de reuniões, umidade externa e do Lara

Conclui-se do gráfico que foi possível controlar a umidade relativa do ambiente com desempenho satisfatório visto que os ambientes vizinhos, no caso o ar externo e o do Lara, se encontram com umidades fora da zona desejável. É importante ressaltar novamente que os resultados foram obtidos em um período de chuvas; neste caso, entretanto, esse fator foi benéfico para a climatização visto que há sempre a captação do ar externo.

A Figura 5.11 mostra a mesma relação entre as umidades relativas porém para o segundo experimento entre os dias 1 e 2 Dezembro onde se aplicou a segunda estratégia de controle.

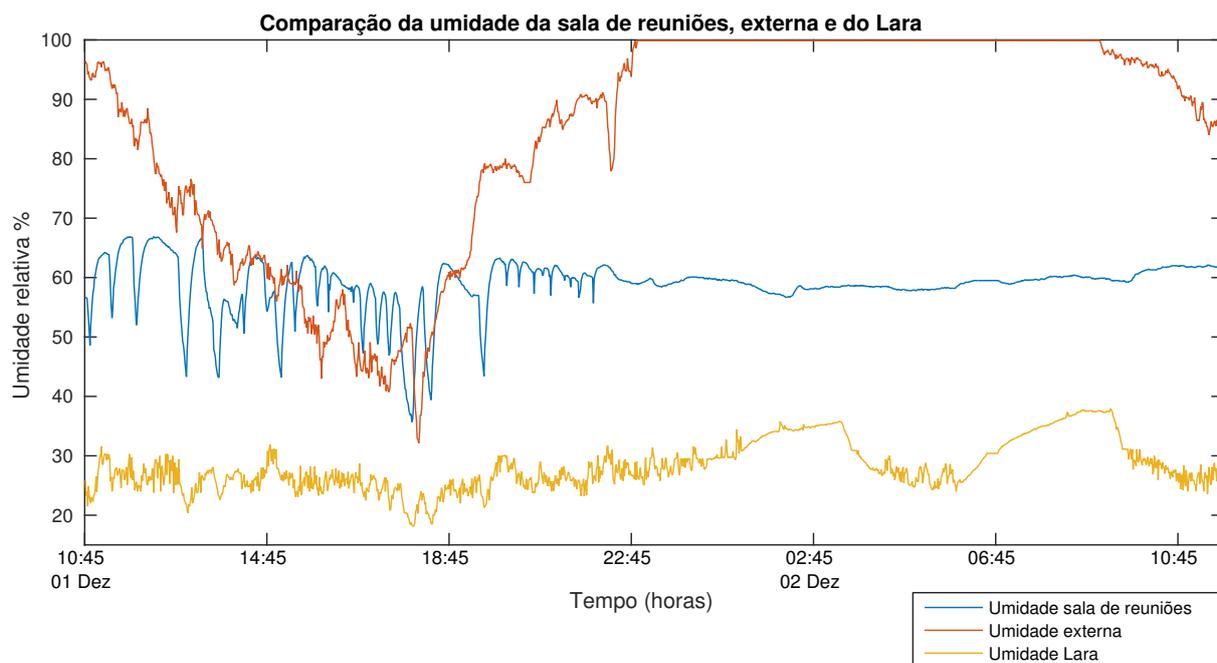


Figura 5.11: Comparação da umidade interna da sala de reuniões, umidade externa e do Lara, experimento 2

Nota-se do gráfico que a umidade relativa da sala de reuniões novamente se manteve dentro da faixa esperada, mesmo com maior utilização da bomba d'água e menor utilização do compressor, como mostrado na Seção 5.1. O gráfico mostra também que mesmo em períodos do dia onde tanto a umidade externa quanto a da sala vizinha (Lara) estavam em um nível baixo o sistema foi capaz de manter a sala de reuniões umidificada.

Capítulo 6

Conclusões

A utilização eficiente de recursos é extremamente importante no contexto desse novo paradigma em que a humanidade se encontra. O conceito de sustentabilidade aliado ao conforto e praticidade buscados pelo ser humano são os motivadores da automação predial e conseqüentemente de ambientes inteligentes.

Dessa forma, o sistema de climatização híbrida se mostrou um recurso viável na busca de conforto térmico aliado à um gasto energético responsável onde uma implementação de um sistema de automação robusto pode trazer vários benefícios. Pode-se ver que a implementação de uma estratégia de controle compatível com o sistema pode trazer ganhos em relação ao consumo sem perda no que tange a refrigeração e umidificação do ambiente. Uma redução superior à 40% pode ser atingida com esses sistemas se projetados de maneira correta e com uma manutenção constante.

O controle TITO proposto neste trabalho teve resultados satisfatórios, mesmo que levando em consideração o modo convencional como o modo principal e o evaporativo como secundário. Seria interessante a implementação e teste do controle em um período de seca onde o sistema evaporativo se mostra mais eficiente e portanto o ganho energético pode ser ainda melhor.

Devido à problemas na conciliação de horários, dificuldades com a rede elétrica e outros não foi possível se realizar mais testes para obtenção de mais dados para análise; porém os resultado podem ser considerados satisfatórios.

6.1 Perspectivas Futuras

Alguns trabalhos podem ser feitos para melhorar o sistema híbrido do Lara em especial

- Manutenção do sistema híbrido como um todo e *retrofitting* visto que o rendimento está abaixo do esperado
- Implementação de um controle para a velocidade do ventilador e variação da captação do ar externo.
- Projeto de controle adaptado voltado a detecção de movimento na sala de reuniões levando em conta o fluxo e movimentação de pessoas.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] LAMBERTS, R. et al. Conforto e stress térmico. *LabEEE - Universidade Federal de Santa Catarina*, 2011.
- [2] LOPES, R. P.; SILVA, J. de Sousa e; REZENDE, R. C. Princípios básicos de psicometria. In: _____. [S.l.]: Universidade Federal de Viçosa, 2008. cap. 3.
- [3] <https://disciplinas.stoa.usp.br>.
- [4] CAMARGO, J. R. Resfriamento evaporativo: poupando a energia e o meio ambiente. *Rev. ciênc. exatas, Taubaté*, v. 9/10, n. 1-2, p. 69-75, 2004.
- [5] VIANNA, P. I. E. Automação de um sistema de climatização híbrido com foco no processo evaporativo visando conforto térmico e uso racional da energia. *Departamento de Engenharia Elétrica, Universidade de Brasília*, 2012.
- [6] OLMOS, J. L. *Sistema Híbrido de Climatização Visando Conforto Térmico e Eficiência Energética*. Tese (Dissertação de Mestrado em Engenharia Elétrica) — Departamento de Engenharia Elétrica, Universidade de Brasília, 2009.
- [7] QUIRINO, P. P. D. S. *Estratégias de Controle Aplicadas a Reatores de Polimerização de Eteno em Solução*. Tese (Dissertação de Mestrado) — Universidade Federal da Bahia, 2012.
- [8] BOAVENTURA, D. B. Automação com openhab de sistema de climatização híbrida visando eficiência energética. *Universidade de Brasília*, 2016.
- [9] ARAÚJO, H. A. S.; MELO, M. C. C. Controle adaptativo da climatização predial para eficiência energética. *Faculdade de Tecnologia - Universidade de Brasília*, 2013.
- [10] HENSEN, J. L. M. *On the thermal interaction of building structure and heating and ventilating system*. Tese (Doutorado) — Technische Universiteit Eindhoven, 1991.
- [11] ANSI/ASHRAE Standard 55 Thermal Environment Conditions for Human Occupancy. 2004.
- [12] DJONGYANG, N.; TCHINDA, R.; NJOMO, D. Thermal comfort: A review paper. *Renewable and Sustainable Energy Reviews*, 2010.
- [13] FANGER, P. O. Thermal comfort, analysis and application in environmental engineering. *Copenhagen: Danish Technical Press*, 1970.

- [14] DOSSAT, R. *Manual de Refrigeração*. 1a edição. ed. São Paulo, Brasil, 1980.
- [15] STULL, R. Wet-bulb temperature from relative humidity and air temperature. *University of British Columbia, Vancouver, British Columbia, Canada*, 2011.
- [16] MAZOCO, B. M.; FILHO, J. D.; MATTEDI, A. Identificação e desacoplamento de malhas e sintonia de controladores de uma planta de queima de usina de pelotização. *Universidade Federal do Espírito Santo*.
- [17] MACEDO, A. M.; WIIRA, M. C. de F. Estudo de técnicas de controle aplicadas a uma bancada didática de quatro tanques. *Universidade de Brasília*, 2015.
- [18] PAULO, T. F. *Controle Adaptativo com Desacoplamento Aplicado a um Sistema de Tanques Acoplados MIMO*. Tese (Dissertação de Mestrado em Engenharia Elétrica) — Universidade Federal do Rio Grande do Norte, 2015.
- [19] LUYBEN, W. L. Distillation decoupling. *AIChE Journal*, v. 16(2), p. 198–203, 1970.
- [20] <http://www.basenge.com.br/tabela-reducao-temperatura-sistema-resfriamento-eva\porativo>.

ANEXOS

1. Especificações do sensor DHT22
2. Código Matlab simulação
3. Código Arduino

Tabela 1: Sensor DHT22 (ccspicc.blospot.com)

Parameters	Conditions	Minimum	Typical	Maximum
Humidity				
Resolution		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			± 1%RH	
Accuracy	25°C		± 4%RH	
	0-50°C			± 5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0°C	30%RH		90%RH
	25°C	20%RH		90%RH
	50°C	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25°C , 1m/s Air	6 S	10 S	15 S
Hysteresis			± 1%RH	
Long-Term Stability	Typical		± 1%RH/year	
Temperature				
Resolution		1°C	1°C	1°C
			8 Bit	8 Bit
Repeatability			± 1°C	
Accuracy		± 1°C		± 2°C
Measurement Range		0°C		50°C
Response Time (Seconds)	1/e(63%)	6 S		30 S

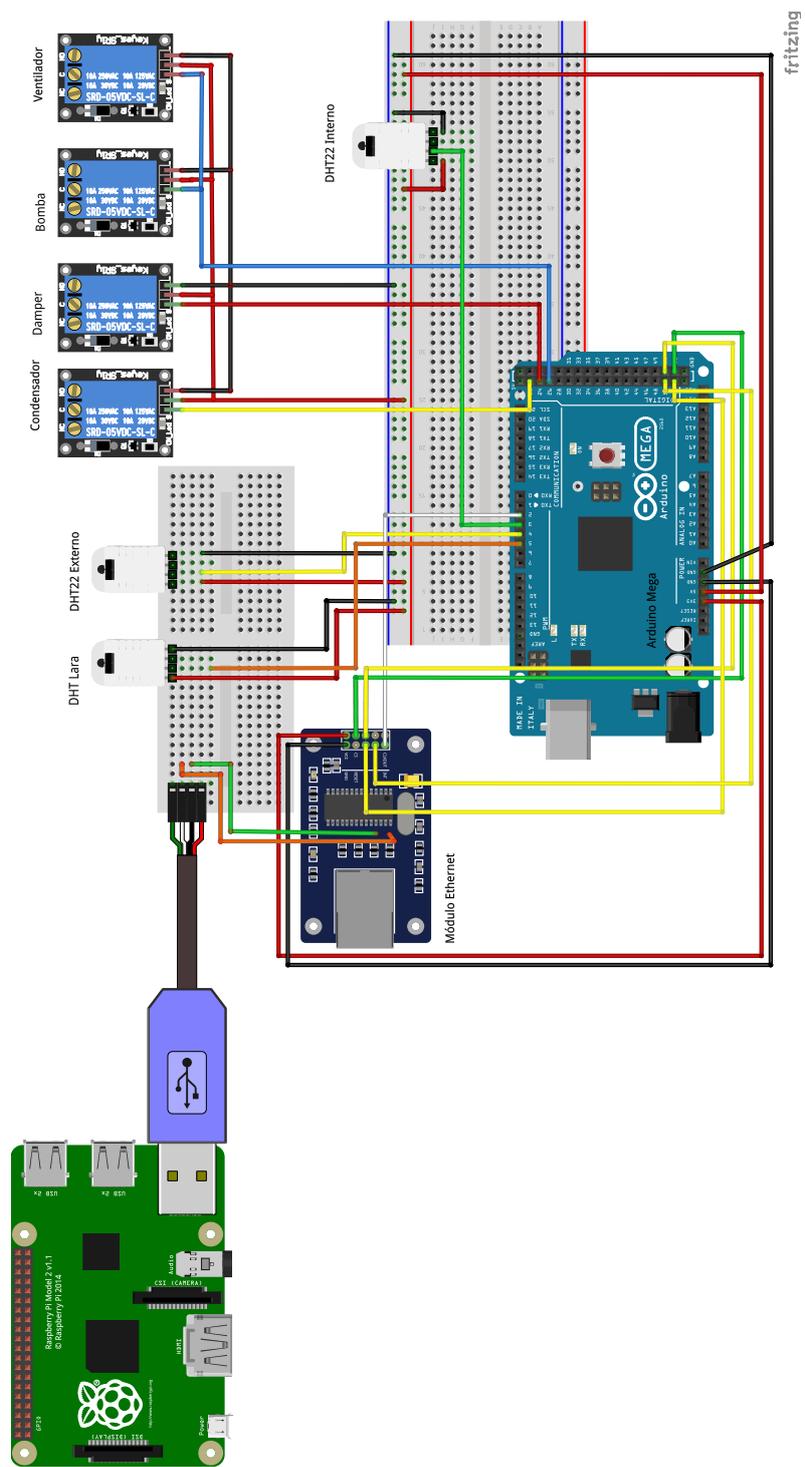


Figura A.I: Esquemático da conexão

I. PROGRAMAS UTILIZADOS

I.1 Programas Matlab

```
1 %% CARREGAR OS DADOS
2 clc;
3 A = 6.112;
4 m = 17.62;
5 Tn = 243.12;
6 C = 2.1674;
7 f= 1.0047;
8 T1 = '2016-10-19 10:00:00.0';
9 T2 = '2016-10-21 10:00:00.0';
10
11 %getmysql;
12 load('dados_sala');
13 s_time=1/60;
14
15 index1 = find(strcmp({t_externa.Time{:}}, T1)==1)
16 index2 = find(strcmp({t_externa.Time{:}}, T2)==1)
17 t_external.signals.values = t_externa.Value(index1:index2);
18 t_external.time=(1:length(t_external.signals.values));
19 t_external.signals.dimensions=1;
20 tempo = length(t_external.signals.values);
21
22 index1 = find(strcmp({u_externa.Time{:}}, T1)==1)
23 index2 = find(strcmp({u_externa.Time{:}}, T2)==1)
24 u_external.signals.values= u_externa.Value(index1:index2);
25 u_external.time = (1:length(u_external.signals.values));
26 u_external.signals.dimensions=1;
27
28 index1 = find(strcmp({t_lara.Time{:}}, T1)==1)
29 index2 = find(strcmp({t_lara.Time{:}}, T2)==1)
30 t_vizinhanca.signals.values = t_lara.Value(index1:index2);
31 t_vizinhanca.time = (1:length(t_vizinhanca.signals.values));
32 t_vizinhanca.signals.dimensions = 1;
33
34 index1 = find(strcmp({u_lara.Time{:}}, T1)==1)
35 index2 = find(strcmp({u_lara.Time{:}}, T2)==1)
36 u_vizinhanca.signals.values = u_lara.Value(index1:index2);
37 u_vizinhanca.time = (1:length(u_vizinhanca.signals.values));
38 u_vizinhanca.signals.dimensions=1;
39
40 index1 = find(strcmp({bomba.Time{:}}, T1)==1)
41 index2 = find(strcmp({bomba.Time{:}}, T2)==1)
42 t_bomba.signals.values = bomba.Value(index1:index2);
43 t_bomba.time = (1:length(t_bomba.signals.values));
44 t_bomba.signals.dimensions = 1;
45
46 index1 = find(strcmp({condensador.Time{:}}, T1)==1)
47 index2 = find(strcmp({condensador.Time{:}}, T2)==1)
48 t_condensador.signals.values = condensador.Value(index1:index2);
49 t_condensador.time = (1:length(t_condensador.signals.values));
50 t_condensador.signals.dimensions = 1;
51
52 index1 = find(strcmp({t_interna.Time{:}}, T1)==1)
53 index2 = find(strcmp({t_interna.Time{:}}, T2)==1)
54 t_sala.signals.values = t_interna.Value(index1:index2);
55 t_sala.time = (1:length(t_sala.signals.values));
56 t_sala.signals.dimensions = 1;
57
58 index1 = find(strcmp({u_interna.Time{:}}, T1)==1)
59 index2 = find(strcmp({u_interna.Time{:}}, T2)==1)
60 u_sala.signals.values = u_interna.Value(index1:index2);
61 u_sala.time = (1:length(u_sala.signals.values));
62 u_sala.signals.dimensions = 1;
63
64 index1 = find(strcmp({tbu_interna.Time{:}}, T1)==1)
65 index2 = find(strcmp({tbu_interna.Time{:}}, T2)==1)
66 tbu_sala.signals.values = u_interna.Value(index1:index2);
67 tbu_sala.time = (1:length(u_sala.signals.values));
68 tbu_sala.signals.dimensions = 1;
69
70 index1 = find(strcmp({tbu_externa.Time{:}}, T1)==1)
```

```

71 index2 = find(strcmp({tbu_externa.Time{:}}, T2)==1)
72 tbu_external.signals.values = u_interna.Value(index1:index2);
73 tbu_external.time = (1:length(u_sala.signals.values));
74 tbu_external.signals.dimensions = 1;
75
76 %% Ganhos do modelo
77 fator = 60;
78
79 K = 0.000222 * fator; % SALA
80 Kr = 0.016835* fator;
81 Kv = 0.75959 * fator; % LAVSII
82 Ke = 0.395 * fator;
83
84 Kc = 1 * fator;
85 Kv = 0.647 * fator;
86
87 %Constantes de tempo
88 %ao = 1;
89 ar = 1489 / fator;
90 al = 574.81 / fator;
91 av = 414.17 / fator;
92 ae = 329.2 / fator;
93 ac = 370.96 / fator;
94
95 %Constantes entre malhas
96 Kut = 0;
97 Ktu = 0.03;
98
99 % parametros estimados pelo idgrey
100 aue = 80.8223 / fator;
101 aum = 171.6923 / fator;
102 Kue = 0.1513;
103 Kum = -1.0134e-3;
104
105 au = 168.5014 / fator;
106 auv = 164.9597 / fator;
107 Ku = 2.1905;
108 Ks = 2.9186e-2;
109 Kuv = 1.5768;
110 Kut = 0.05;
111 Ktu = 0.04;
112
113 %% simulação
114
115 sim('sala2')
116
117 %% plotar resultados
118 close all;
119 s_time=1;
120 plot(t_est.time*s_time,t_est.signals(1).values(:,1));
121 hold on;
122 plot(t_est.time*s_time,t_est.signals(1).values(:,2));
123 xlabel 'Tempo (min)';
124 ylabel 'Temperatura (\^oC)';
125 legend ('Estimado','Real');
126
127 figure(2);
128 plot(u_est.time*s_time,u_est.signals(1).values(:,1));
129 hold on;
130 plot(u_est.time*s_time,u_est.signals(1).values(:,2));
131 xlabel 'Tempo (min)';
132 ylabel 'Umidade Relativa (\%)';
133 legend ('Estimado','Real');
134 title 'Modelo Umidade: Teorico x Real';

```

I.2 Programa arduino

```
1 #include <Dns.h>
2 #include <Dhcp.h>
3 #include <UIPClient.h>
4 #include <UIPServer.h>
5 #include <UIPUDP.h>
6 #include <UIPEthernet.h>
7 #include <ethernet_comp.h>
8 #include <DHT.h>
9 #include <PubSubClient.h>
10 #include <UIPEthernet.h>
11 #include "PID_v1.h"
12
13 #define condensador 22
14 #define damper 24
15 #define bomba 26
16
17 #define sensorReuniao 3
18 #define sensorExterno 4
19 #define sensorLARA 5
20 #define intervaloSensores 5000
21
22 DHT dht (sensorReuniao , DHT22);
23 DHT dht2(sensorExterno , DHT22);
24 DHT dht3(sensorLARA , DHT22);
25
26
27 bool forcandoLeitura = true;
28 bool converteCtoFahr = false;
29 bool compressor = false;
30 bool b_agua = false;
31
32 unsigned long t_medicao;
33 unsigned long tempo = 0;
34
35 double KP = 1.2;
36 double KI = 0.205;
37 double KD = 0;
38 double u_controle, t, currentWindowPidOutput = 0, setpoint_T= 25.0;
39 const double IntThresh = 100;
40 char message_buff[100];
41 char u_contr[10];
42 unsigned long windowStartTime, windowSize = 80000;
43
44 // Parmetros de Internet
45
46 byte ArduinoMAC[] = { 0xDE, 0xED, 0xBA, 0xFE, 0xFE, 0xED };
47 byte Arduinoip[] = { 192, 168, 1, 101 };
48 byte ServerIP[] = { 192, 168, 1, 4 };
49 int ServerPort = 1883;
50
51 EthernetClient ethClient;
52 PubSubClient client(ethClient);
53
54
55 void(* resetFunc) (void) = 0;
56 unsigned long int ResetPeriodo = 86400000;
57
58 //Definicao do PID
59 PID myPID(&t, &u_controle, &setpoint_T, KP, KI, KD, REVERSE);
60
61 //Funcao trata eventos do openhab
62 void callback(char* topic, byte* payload, unsigned int length)
63 {
64     int i;
65     if(!(strcmp(topic, "setpointT"))) )
66     {
67         for(i=0; i<length; i++)
68         {
69             message_buff[i] = payload[i];
70         }
71     }
72 }
```

```

71     message_buff[i] = '\0';
72     char *p_payload = message_buff;
73     setpoint_T = strtod(p_payload, NULL);
74 }
75
76 // Se a mensagem recebida for do topico "condensador", ligar ou desligar ...
77 // o compressor
78 else if( !(strcmp(topic, "condensador")) )
79 {
80     if (length == 2)
81     {
82         digitalWrite(condensador, LOW);
83         client.publish("condensadorR", "1");
84         compressor = true;
85     }
86     else if (length == 3)
87     {
88         digitalWrite(condensador, HIGH);
89         client.publish("condensadorR", "0");
90         compressor = false;
91     }
92 }
93
94 // Se a mensagem recebida for do topico "bomba", ligar ou desligar o ...
95 // compressor
96 else if( !(strcmp(topic, "bomba")) )
97 {
98     if (length == 2)
99     {
100         digitalWrite(bomba, LOW);
101         client.publish("bombaR", "1");
102         b_agua = true;
103     }
104     else if (length == 3)
105     {
106         digitalWrite(bomba, HIGH);
107         client.publish("bombaR", "0");
108         b_agua = false;
109     }
110 }
111 // Se a mensagem recebida for do topico "damper", abrir ou fechar o damper
112 else if( !(strcmp(topic, "damper")) )
113 {
114
115     if (length == 2)
116     {
117         digitalWrite(damper, LOW);
118         client.publish("damperR", "1");
119     }
120     else if (length == 3)
121     {
122         digitalWrite(damper, HIGH);
123         client.publish("damperR", "0");
124     }
125 }
126 }
127
128 void setup() {
129
130     Serial.begin(9600);
131
132     // Inicializando os pinos de atuação
133     pinMode(damper, OUTPUT);
134     pinMode(condensador, OUTPUT);
135     pinMode(bomba, OUTPUT);
136     digitalWrite(damper, HIGH);
137     digitalWrite(bomba, HIGH);
138     digitalWrite(condensador, HIGH);
139

```

```

140 // Inicializando os sensores
141 dht.begin(); //dht interno
142 dht2.begin(); //dht externo
143 dht3.begin(); //dht do LARA
144
145 // Inicia a internet
146 client.setServer(ServerIP, ServerPort);
147 client.setCallback(callback);
148 Ethernet.begin(ArduinoMAC, Arduinoip);
149
150 // Coloca o momento atual como atualizacao dos sensores, e atualiza tudo ...
151 // na rede
152 float t = dht.readTemperature(convertCtoFahr, forçandoLeitura);
153 Serial.println(t);
154 t_medicao = millis();
155 if (client.connect("arduinoClient")) {
156     client.subscribe("damper");
157     client.subscribe("condensador");
158     client.subscribe("bomba");
159     client.subscribe("setpointT");
160     client.subscribe("setpointH");
161     client.publish("condensadorR", "0");
162     client.publish("damperR", "0");
163     client.publish("bombaR", "0");
164 }
165 windowStartTime = millis();
166 //dtostrf(1.0,4,1,u_contr);
167 myPID.SetMode(AUTOMATIC);
168 myPID.SetOutputLimits(0, 100);
169 }
170
171 void loop()
172 {
173     // Existem duas situacoes para reiniciar o arduino (mais efetivo que ...
174     // reiniciar a conexao).
175     // Situacao 1 - Se a conexao caiu.
176     // Situacao 2 - Reinicializacao programada depois de um periodo fixo ...
177     // estabelecido la em cima
178     if ( (!client.connected()) || (millis() > ResetPeriodo) ){
179         Serial.print("\nCurr time = ");
180         Serial.print(millis());
181         Serial.print("\n\n **** RESETEI! ****\n");
182         delay(1000);
183         resetFunc();
184     }
185
186     client.loop();
187
188     // t ≤ Δ T = Tempo atual - ultima atualização dos sensores
189     tempo = millis() - t_medicao;
190
191     if (tempo > intervaloSensores) {
192
193         //Lendo os dados dos sensores
194         float h = dht.readHumidity(forçandoLeitura);
195         float h2 = dht2.readHumidity(forçandoLeitura);
196         float h3 = dht3.readHumidity(forçandoLeitura);
197         t = dht.readTemperature(convertCtoFahr, forçandoLeitura);
198         float t2 = dht2.readTemperature(convertCtoFahr, forçandoLeitura);
199         float t3 = dht3.readTemperature(convertCtoFahr, forçandoLeitura);
200
201         // Calculando as variáveis indiretas
202         float tbuIn = t *atan(0.151977*pow((h +8.313659),0.5))+atan(t+h ...
203             )-atan(h -1.676331)+0.00391838*pow(h ,1.5)*atan(0.023101*h )-4.686035;
204         float tbuOut = ...
205             t2*atan(0.151977*pow((h2+8.313659),0.5))+atan(t2+h2)-atan(h2-1.676331)+0.00391838*p
206         float tbuLARA = ...
207             t3*atan(0.151977*pow((h3+8.313659),0.5))+atan(t3+h3)-atan(h3-1.676331)+0.00391838*p

```

```

204 float es = 6.112*exp((17.67*t)/(t+243.5));
205 float e = (es*h)/100;
206 float td = (243.5*log(e/6.112))/(17.67-log(e/6.112));
207
208 // Espaço para a conversão float -> ASCII
209 char umidade_Reuniao[10];
210 char umidade_Externa[10];
211 char umidade_LARA[10];
212 char temperatura_Reuniao[10];
213 char temperatura_Externa[10];
214 char temperatura_LARA[10];
215 char tbu_Externa[10];
216 char tbu_Interna[10];
217 char tbu_LARA[10];
218 char temp_Dew[10];
219
220
221
222 // Convertendo os dados de float para ASCII
223 dtostrf(h,4,1,umidade_Reuniao);
224 dtostrf(t,4,1,temperatura_Reuniao);
225 dtostrf(h2,4,1,umidade_Externa);
226 dtostrf(t2,4,1,temperatura_Externa);
227 dtostrf(h3,4,1,umidade_LARA);
228 dtostrf(t3,4,1,temperatura_LARA);
229 dtostrf(tbuOut,4,1,tbu_Externa);
230 dtostrf(tbuIn,4,1,tbu_Interna);
231 dtostrf(tbuLARA,4,1,tbu_LARA);
232 dtostrf(td,4,1,temp_Dew);
233
234 // Atualizando o tempo dos sensores e verificando na rede o ...
    funcionamento do sistema.
235 t_medicao = millis();
236 client.publish("reuniao/temperatura",temperatura_Reuniao);
237 client.publish("reuniao/umidade",umidade_Reuniao);
238 client.publish("reuniao/tbu",tbu_Interna);
239 client.publish("reuniao/tempdew",temp_Dew);
240 client.publish("externa/temperatura",temperatura_Externa);
241 client.publish("externa/umidade",umidade_Externa);
242 client.publish("externa/tbu",tbu_Externa);
243 client.publish("LARA/temperatura",temperatura_LARA);
244 client.publish("LARA/umidade",umidade_LARA);
245 client.publish("LARA/tbu",tbu_LARA);
246
247 if((tbuIn>tbuOut+2))
248 {
249     evaporativo(1);
250 }
251 else
252 {
253     evaporativo(2);
254 }
255 if(h<40)
256 {
257     evaporativo(1);
258 }
259 else
260 {
261     evaporativo(2);
262 }
263 }
264 delay(1000);
265 myPID.SetSampleTime(1000);
266 myPID.Compute();
267 //Serial.println(u_controle);
268 updateOutput();
269 //Serial.println(setpoint_T);
270 //Serial.println(t);
271
272 }

```

```

273
274 void updateOutput()
275 {
276     unsigned long now = millis();
277     if(now - windowStartTime > windowSize)
278     {
279         //time to shift the window
280         windowStartTime += windowSize;
281         currentWindowPidOutput = u_contrle;
282     }
283
284     if(currentWindowPidOutput * windowSize > ((now - windowStartTime) ))
285     {
286         if(!compressor)
287         {
288             compressor = true;
289             Serial.println("ON");
290             digitalWrite(condensador,LOW);
291             //dtostrf(1.0,4,1,u_contr);
292             client.publish("temperatura/control", "1");
293             client.publish("condensadorR", "1");
294         }
295     }
296     else if(compressor)
297     {
298         compressor = false;
299         Serial.println("OFF");
300         digitalWrite(condensador,HIGH);
301         client.publish("temperatura/control", "0");
302         client.publish("condensadorR", "0");
303     }
304     Serial.print("\n");
305     Serial.println(now);
306     Serial.println(windowStartTime);
307     Serial.println(currentWindowPidOutput);
308 }
309
310
311 void evaporativo(int n)
312 {
313     if (n==1)
314     {
315         digitalWrite(bomba,LOW);
316         client.publish("bombaR", "1");
317         b_agua = true;
318     }
319     else
320     {
321         digitalWrite(bomba,HIGH);
322         client.publish("bombaR", "0");
323         b_agua = false;
324     }
325 }

```