



UNIVERSIDADE FEDERAL DO RIO GRANDE
ESCOLA DE QUÍMICA E ALIMENTOS
PÓS-GRADUAÇÃO EM ENGENHARIA E CIÊNCIA DE ALIMENTOS

**MONITORAMENTO POR MICROCONTROLADOR DO CULTIVO
DE *SPIRULINA* EM FOTOBIORREATOR *RACEWAY***

Renato Dutra Pereira Filho

Prof. Dr. Jorge Alberto Vieira Costa
Orientador

Rio Grande/RS

2014

UNIVERSIDADE FEDERAL DO RIO GRANDE
ESCOLA DE QUÍMICA E ALIMENTOS
PÓS-GRADUAÇÃO EM ENGENHARIA E CIÊNCIA DE ALIMENTOS

**MONITORAMENTO POR MICROCONTROLADOR DO CULTIVO
DE *SPIRULINA* EM FOTOBIORREATOR *RACEWAY***

Renato Dutra Pereira Filho

Tese apresentada como um dos requisitos para obtenção do título de Doutor no Programa de Pós-Graduação em Engenharia e Ciência de Alimentos da Universidade Federal do Rio Grande.

Prof. Dr. Jorge Alberto Vieira Costa
Orientador

Rio Grande/RS

2014

Dedico este trabalho
para Cláudia e Davi,
queridos esposa e filho.

AGRADECIMENTOS

Ao Prof. Dr. Jorge Alberto Vieira Costa, pela receptividade, desde quando conversamos pela primeira vez sobre meu ingresso no curso, pela autonomia, pela amizade, pela acolhida, pelo apoio, pela compreensão e pelo aprendizado.

À minha esposa Cláudia Bonoto Dutra Pereira, por tudo – realmente não tenho palavras para agradecer!

Ao meu filho Davi Bonotto Dutra Pereira, que sentenciou aos 4 anos de idade, em 2009: “vai lá e faz de novo!”.

Aos meus familiares, todos, aos quais fiquei devendo quase cinco anos do meu convívio, por estar ligeiramente ocupado, preparando esta surpresa.

Aos meus pais Renato e Diva, pela criação, pela educação, pela resiliência, pelo exemplo, pelo sacrifício!

À minha irmã Márcia, pela risada farta no momento certo, e por me ensinar a rir de mim mesmo.

Ao amigo Engenheiro Vitor Badiale Furlong, por ser o melhor colaborador que eu poderia ter tido.

Ao Engenheiro Márcio Pessoa, meu padrinho de casamento, colega de Universidade, sobretudo amigo, pelas sugestões, em especial na sua área de conhecimento e pela torcida.

Ao estudante de Engenharia Química Victor Botelho, pela disposição de se oferecer para trabalhar, mesmo sem bolsa.

À Profa. Dra. Eliana Badiale Furlong e ao Prof. Dr. Luís Antônio de Almeida Pinto, por acolherem a minha proposta de cursar o Doutorado, ao fazerem parte da banca de seleção.

Aos Professores dos Núcleos de Engenharia Química e de Engenharia de Alimentos da Escola de Química e Alimentos da FURG, aqueles engajados e colaborativos em prol do que fazem, em especial, dos cursos de graduação em Engenharia mantidos pela EQA.

Ao colega Professor Titular Henrique da Costa Bernardelli, não só pela companhia, amizade e cafezinhos, mas pela sugestão de que eu fizesse o curso e de que eu procurasse o Prof. Jorge.

Aos Professores Dra. Christiane Saraiva Ogradowski e Dr. Fabrício Butierres Santana pelo incentivo, pelas sugestões, pelo empréstimo da impressora 3D, mas especialmente pelo companheirismo.

Aos Professores do Programa de Pós-Graduação em Engenharia e Ciência de Alimentos (PPGECA), por acreditarem sempre, desde a concepção original, que a FURG poderia ter um curso como este, de qualidade e relevância. Sem a fé e o esforço de vocês, nós estudantes não teríamos a oportunidade de realizar nosso doutoramento aqui. Muito Obrigado!

Aos colegas de pós-graduação, todos aqueles com quem fiz créditos em disciplinas, ao voarmos juntos, muitas vezes diminuimos a resistência do ar uns para os outros.

Aos colegas Bruno Ladeira, José Tavares e Paulo Eduardo Hansmann, pela amizade e companhia no transcorrer das disciplinas do curso.

A todos os colegas de LEB, pela possibilidade de realizarmos essa viagem juntos.

Ao Dr. Adriano H. Arruda e à Dra. Ana Cláudia Margarites, pelo auxílio nas horas incertas.

Ao Prof. Dr. Otavio Socolowski Júnior, pelas sugestões de bibliografia sobre a interação da radiação luminosa com a matéria.

À Universidade Federal do Rio Grande e ao Programa de Pós Graduação em Engenharia e Ciência de Alimentos, pela oportunidade.

Ao Engenheiro Marcel Mougenot, por me perguntar pela plataforma Arduino, em 2011, ao final de uma aula de Instrumentação e Controle de Processos Químicos e me fazer pesquisar a respeito da mesma, pois dela nada, até então, eu sabia.

A todos os estudantes das disciplinas que ministrei à noite, pela compreensão de que aquele era o horário possível, para que eu pudesse me qualificar.

A todos os meus professores, ao longo de toda a minha vida, mas especialmente para D. Terezinha, D. Luna, D. Vilma, D. Sueli, D. Roma, D. Neusa e aquelas que eu não lembrei, professoras dos então “Grupos Escolares” em que cursei desde o Jardim de Infância até a 5ª série, **Fernando Caruccio e Fernando Osório**, por terem colaborado em muito não só para a minha formação, mas de tantos como eu!

A Deus, pela força durante as provações e pela luz em busca da verdade!

“...Some things you should let go, there only gonna pull you down,

Just like weight on your shoulder

They are only gonna make you drown

We all swing high, we all swing low,

We all got secrets people don't know

We all got dreams we can't let go,

We want to brave, Don't be afraid

Your heart is too heavy from things you carry a long time,

You been up you been down, tired and you don't know why,

But you're never gonna go back, you only live one life

Let go, let go, let gooooo, Let go, let go, let go...”

Matshyahu, “Live Like a Warrior”

MONITORAMENTO POR MICROCONTROLADOR DO CULTIVO DE *SPIRULINA* EM FOTOBIORREATOR RACEWAY

RESUMO

A oportunidade de produção de biomassa microalgal tem despertado interesse pelos diversos destinos que a mesma pode ter, seja na produção de bioenergia, como fonte de alimento ou servindo como produto da biofixação de dióxido de carbono. Em geral, a produção em larga escala de cianobactérias e microalgas é feita com acompanhamento através de análises físico-químicas *offline*. Neste contexto, o objetivo deste trabalho foi monitorar a concentração celular em fotobiorreator *raceway* para produção de biomassa microalgal usando técnicas de aquisição digital de dados e controle de processos, pela aquisição de dados *inline* de iluminância, concentração de biomassa, temperatura e pH. Para tal fim foi necessário construir sensor baseado em software capaz de determinar a concentração de biomassa microalgal a partir de medidas ópticas de intensidade de radiação monocromática espalhada e desenvolver modelo matemático para a produção da biomassa microalgal no microcontrolador, utilizando algoritmo de computação natural no ajuste do modelo. Foi projetado, construído e testado durante cultivos de *Spirulina* sp. LEB 18, em escala piloto *outdoor*, um sistema autônomo de registro de informações advindas do cultivo. Foi testado um sensor de concentração de biomassa baseado na medição da radiação passante. Em uma segunda etapa foi concebido, construído e testado um sensor óptico de concentração de biomassa de *Spirulina* sp. LEB 18 baseado na medição da intensidade da radiação que sofre espalhamento pela suspensão da cianobactéria, em experimento no laboratório, sob condições controladas de luminosidade, temperatura e fluxo de suspensão de biomassa. A partir das medidas de espalhamento da radiação luminosa, foi construído um sistema de inferência *neurofuzzy*, que serve como um sensor por software da concentração de biomassa em cultivo. Por fim, a partir das concentrações de biomassa de cultivo, ao longo do tempo, foi prospectado o uso da plataforma Arduino na modelagem empírica da cinética de crescimento, usando a Equação de Verhulst. As medidas realizadas no sensor óptico baseado na medida da intensidade da radiação monocromática passante através da suspensão, usado em condições *outdoor*, apresentaram baixa correlação entre a concentração de biomassa e a radiação, mesmo para concentrações abaixo de 0,6 g/L. Quando da investigação do espalhamento óptico pela suspensão do cultivo, para os ângulos de 45° e 90° a radiação monocromática em 530 nm apresentou um comportamento linear crescente com a concentração, apresentando coeficiente de determinação, nos dois casos, 0,95. Foi possível construir um sensor de concentração de biomassa baseado em software, usando as informações combinadas de intensidade de radiação espalhada nos ângulos de 45° e 135° com coeficiente de determinação de 0,99. É factível realizar simultaneamente a determinação *inline* de variáveis do processo de cultivo de *Spirulina* e a modelagem cinética empírica do crescimento do micro-organismo através da equação de Verhulst, em microcontrolador Arduino.

Palavras-chave: *datalogging*, espalhamento da luz, lógica nebulosa, microcontrolador Arduino, otimização por enxame de partículas, *soft sensing*

SPIRULINA CULTURE MICROCONTROLLED MONITORING WITHIN RACEWAY PHOTOBIOREACTOR

ABSTRACT

Increasing interest has been given to microalgae due to the wide range of destinations where the produced biomass may be applied. Ranging from food products, bioenergy production, or using the culture itself as a carbon dioxide biofixation process. Usually, cyanobacteria and microalgae large scale production physical-chemical analysis are analyzed through offline methodologies. Within this context, this work objective was to monitor microalgae cellular concentration automatically within raceway photobioreactor using digital data acquisition and process control through inline acquisition of luminance data, biomass concentration, temperature and pH. To achieve such, a software based sensor capable of determining microalgae biomass concentration from scattered monochromatic radiation and mathematical model with natural computing parameters search was constructed. The instrument was tested thoroughly in several outdoor pilot scale *Spirulina sp* LEB 18 cultures. Alongside the instrument, an automated culture data logger was also designed. The first biomass sensor based its principle in emitting light through the culture and measuring passing radiation. Thereafter, a second array of sensors were constructed. In the later array, the measured radiation was the portion scattered by the cyanobacteria from its original path. Both instruments were calibrated under controlled luminosity, temperature and biomass flow conditions. The light scattering data enable a neurofuzzy inference system in order to convert such data into biomass concentration within the culture. Ultimately, from the culture biomass data, the utilization of the Arduino platform was investigated in order to realize empirical modeling of the growth kinetics using the Verhulst equation. The non-scattered monochromatic light biomass sensor, when applied to outdoor conditions, achieved poor correlations among biomass concentration and measured radiation, for biomass concentrations smaller than 0.6 g.L^{-1} . The scattered light sensor obtained a crescent linear behavior between biomass concentration and measured radiation when using a 530nm emitter and 45° and 90° receivers. In this situation, the coefficient of determination was higher than 0.95. Thus, a software based sensor was constructed using the combined measurements of scattered radiation in the 45° and 135° angles. The coefficient of determination the entire system is approximately 0.99. Therefore, it is feasible to imply that inline determination of process variables and kinetic modeling within the *Spirulina* culture is possible using the Arduino platform.

Keywords: Arduino microcontroller, datalogging, fuzzy logic, light scattering, particle swarm optimization, soft sensing

LISTA DE FIGURAS

CAPÍTULO II - REVISÃO DA LITERATURA

Figura 1 – Exemplo de Topologia de um Sistema ANFIS Genérico (adaptado de Bonissone, 2002).....	29
--	----

CAPÍTULO III - DESENVOLVIMENTO DO TRABALHO

ARTIGO I

Figura 1 - Placa do Arduino Nano v3.0. (a) Portas Analógicas. (b) Portas Digitais.(c) Porta micro USB.....	46
---	----

Figura 2 - (a) Arranjo do Sistema de Monitoramento no <i>Raceway</i> ; (b) Vista Geral do Protótipo de Monitoramento; (c) Ligações do <i>MARK</i> ao Emissor e Sensores.....	48
---	----

Figura 3 – Registro de Temperaturas ao Longo do Cultivo de <i>Spirulina</i> na Estufa <i>Outdoor</i> pelo <i>MARK 2.5</i>	49
--	----

Figura 4 - Conjunto de Medições Realizadas de Iluminância no Interior da Estufa (azul) e no interior do cultivo de <i>Spirulina</i> sp. LEB 18 (vermelho).....	49
---	----

Figura 5 - Evolução da Razão entre a Iluminância no Interior do Cultivo de <i>Spirulina</i> sp. LEB 18 e no Interior da Estufa.....	50
--	----

Figura 6 - Sinal do Receptor Óptico de Biomassa no Início do Período Noturno e Concentração de Biomassa.....	51
---	----

Figura 7 - Organização Interna dos Componentes do Protótipo <i>MARK 3.0</i> – (a) fonte de 6V; (b) Arduino Nano; (c) Leitor de Cartão SD; (d) Relé; (e) <i>Shield</i> de pH; (f) Relógio de Tempo Real (RTC); (g) Conexões.....	52
--	----

Figura 8 Registro pelo <i>MARK 3.0</i> do pH ao Longo de Cultivo <i>Outdoor</i> de <i>Spirulina</i>	52
--	----

ARTIGO II–

Figura 1 - Equipamento Usado para Realizar as Medidas de Espalhamento da Luz pela Suspensão de Microalga <i>Spirulina</i>	60
--	----

Figura 2 – (a) Projeto da Peça;(b) Suporte Pronto com Emissores e Receptores..61

Figura 3 – (a) Sinal dos Sensores; (b) Responsividade Espectral (TAOS INC. – 2007).....61

Figura 4 – Aspecto do Espalhamento da Luz na Suspensão, (a) em 530 nm e (b) em 660 nm.....63

Figura 5 – Sinal no Sensor TAOS e Concentrações de Biomassa- (a) 135°, (b) 45° e (c) 90°.....67

ARTIGO III

Figura 1 – Arquitetura do Sistema ANFIS.....77

Figura 2 – Algoritmo da Otimização por Enxame de Partículas.....78

Figura 3 – Somatório do Erro Absoluto ao Longo das Iterações Durante o Treinamento da Rede ANFIS – TSK – PSO para os Pares Combinados de Ângulos.....82

4 – Ajuste Após as Iterações e Treinamento da Rede ANFIS – TSK – PSO para os Pares Combinados de Ângulos – Conc. Experimental (Xexp) e do Modelo (Xmod).....83

Figura 5 – Concepção do Sensor de Biomassa Óptico Combinado.....84

ARTIGO IV

Figura 1 – (A) Arduino Nano, (B) Arduino Mega e (C) Arduino Due.....92

Figura 2 – Concentrações de Biomassa – Experimental (Xn) e Determinadas pelo Ajuste dos Modelos nas Placas Arduino (X_Due, X_Mega, X_Nano).....95

Figura 3 – Minimização do Somatório do Erro Quadrático ao Longo das Iterações no Ajuste dos Modelos nas Placas Arduino.....96

LISTA DE TABELAS

CAPÍTULO III - DESENVOLVIMENTO DO TRABALHO

ARTIGO II

Tabela 1 - Sinal no Sensor Luminoso para 530 nm em Função da Concentração e do Ângulo entre Emissor e Receptor (Médias e Desvios padrões de 10000 Amostragens).....65

Tabela 2 - Sinal no Sensor Luminoso para 530 nm em Função da Concentração e do Ângulo entre Emissor e Receptor (Médias e Desvios Padrões de 2500 Amostragens).....66

ARTIGO III

Tabela 1 - Sinais dos Sensores Ópticos em Função da Concentração e do Ângulo entre Emissor e Receptor, Usados no Treinamento do Sensor por Software (ANFIS).....79

Tabela 2 - Parâmetros do Algoritmo de Otimização por Enxame de Partículas (PSO).....79

Tabela 3 - Avaliação dos Parâmetros de Representatividade do Modelo ANFIS+PSO com Funções de Pertinência Lineares Iguamente Espaçadas Após 500 Iterações.....80

ARTIGO IV

Tabela 1 – Principais Características das Placas Arduino92

Tabela 2 - Tempos de Execução do Programa de Modelagem Cinética94

Tabela 3 - Coeficiente de Determinação, R^2 , do Modelo Cinético Logístico de Verhulst95

SUMÁRIO

CAPÍTULO I.....	15
1 INTRODUÇÃO	16
2 OBJETIVOS	17
2.1 Objetivo geral.....	17
2.2 Objetivos específicos	17
3 JUSTIFICATIVA.....	18
3.1 Histórico.....	18
CAPÍTULO II	21
4 REVISÃO DA LITERATURA	22
4.1 Biomassa Microalgal	22
4.2 Biorreatores para Produção de Biomassa Microalgal.....	23
4.3 Natureza Dinâmica do Cultivo de Microalgas.....	25
4.4 Fundamentos da Lógica <i>Fuzzy</i> ou <i>Nebulosa</i>	27
4.5 A Plataforma de Hardware Arduino na Construção de <i>Dataloggers</i>	30
4.6 Sensores Ópticos de Concentração de Biomassa Microalgal.....	33
4.7 A Otimização por Enxame de Partículas.....	35
4.8 Sensores por Software.....	38
CAPÍTULO III.....	40
5 DESENVOLVIMENTO DO TRABALHO.....	41
ARTIGO I: Sistema Autônomo de Monitoramento do Cultivo de Biomassa Microalgal: Desenvolvimento e Teste.....	42
RESUMO	42
ABSTRACT	43
1. INTRODUÇÃO	44
2. MATERIAIS E MÉTODOS	45
3. RESULTADOS E DISCUSSÃO	47
4. CONCLUSÕES	53

5. REFERÊNCIAS	53
ARTIGO II: Efeito da Concentração de <i>Spirulina</i> sp. LEB 18 no Espalhamento da Radiação Luminosa Monocromática.....	55
RESUMO	55
ABSTRACT	56
1 INTRODUÇÃO.....	57
2 MATERIAIS E MÉTODOS.....	59
3 RESULTADOS E DISCUSSÃO	63
4 CONCLUSÕES	68
5 REFERÊNCIAS	69
ARTIGO III: Sensor Óptico Combinado Definido por Software para Determinação da Concentração de <i>Spirulina</i> sp. LEB 18.....	72
RESUMO	72
ABSTRACT	73
1 INTRODUÇÃO.....	74
2 MATERIAIS E MÉTODOS.....	76
3 RESULTADOS E DISCUSSÃO	78
4 CONCLUSÕES	84
5 REFERÊNCIAS	85
ARTIGO IV: Microcontroladores Arduino na Modelagem Cinética de Bioprocessos Autotróficos: Desenvolvimento e Desempenho.....	87
RESUMO	87
ABSTRACT	88
1 INTRODUÇÃO.....	89
2 MATERIAIS E MÉTODOS.....	91
3 RESULTADOS E DISCUSSÃO	93
4 CONCLUSÕES	97
5 REFERÊNCIAS	98
CAPÍTULO IV: CONCLUSÕES GERAIS	100
6 CONCLUSÕES GERAIS	101
7 SUGESTÕES PARA TRABALHOS FUTUROS.....	103

CAPÍTULO V: REFERÊNCIAS	104
REFERÊNCIAS	105
APÊNDICE A - Fotografias das Montagens e Protótipos dos MARKs	115
APÊNDICE B - Código do Programa do Microcontrolador Arduino - MARK 3.0	119
APÊNDICE C - Programa do Arduino Due - para Medição da Intensidade da Radiação Monocromática Espalhada (<i>Scattering</i>) pela Suspensão de Spirulina.....	146
APÊNDICE D - Programa em Scilab 5.4 da Rede ANFIS TSK + PSO	154
APÊNDICE E - Programa de Modelagem Cinética Verhulst - PSO para ARDUINO.....	170

CAPÍTULO I

1 INTRODUÇÃO

O desenvolvimento da automação de processos vem ao encontro das necessidades crescentes de produtividade, qualidade, competitividade e segurança na operação de plantas industriais de processamento de alimentos.

Com o aumento da demanda por produtos alimentícios derivados de microalgas, aumenta também a necessidade de produzir biomassa microalgal. Conforme vão se difundindo a ciência e a tecnologia dos fotobiorreatores para a produção de microalgas, surgem os problemas de engenharia associados ao problema de monitoramento da operação desses fotobiorreatores.

São de vital importância para a automação de fotobiorreatores os instrumentos de medição de concentração de biomassa, para que sejam factíveis as iniciativas de desenvolvimento de aplicação de monitoramento *inline* de processos de cultivo e de produção de biomassa microalgal.

Em geral, na produção de microalgas e cianobactérias em larga escala, há o acompanhamento *offline* do cultivo. Isso caracteriza um grau baixo de monitoramento e de registro sistemático das variáveis de processo de forma automatizada.

O ferramental necessário para monitorar o cultivo de microalgas e cianobactérias deve ser resistente às condições de operação da unidade e propiciar o registro de informações das variáveis de processo, bem como ser capaz de processá-las. Esse conjunto composto pelos sensores, fiação, cabeamento, alimentação elétrica e unidade de gerenciamento deve ter, preferencialmente, sua concepção, projeto e tecnologia conhecidos e acessíveis pelos interessados na operação da planta de produção microalgal.

A vantagem em trabalhar com uma plataforma de instrumentação analítica de processo que seja aberta não só reside no custo, mas na capacidade de operá-la sem a dependência de serviços de terceiros, com tecnologia própria.

2 OBJETIVOS

2.1 Objetivo geral

Monitorar o cultivo de *Spirulina* em fotobiorreator *raceway* para produção de biomassa usando técnicas de aquisição digital de dados e controle de processos.

2.2 Objetivos específicos

Utilizar a aquisição de dados *inline* para monitorar a iluminância, a temperatura e o pH durante a operação de fotobiorreator *raceway*;

Construir sensor baseado em software, usando rede *neurofuzzy*, capaz de determinar a concentração de biomassa microalgal a partir de medidas ópticas;

Desenvolver modelo matemático para a produção da biomassa microalgal, através da computação em microcontrolador;

Utilizar o algoritmo da otimização por enxame de partículas no ajuste do modelo cinético de cultivo desenvolvido.

3 JUSTIFICATIVA

O acompanhamento dos cultivos é realizado normalmente através de análises *offline*, o que implica em retirada de amostras e análises posteriores. Além disso, o intervalo entre as amostragens é aquele factível para os colaboradores poderem realizar o conjunto de atividades associadas com o acompanhamento do bioprocessamento.

A possibilidade de monitorar o cultivo em fotobiorreator do tipo *raceway* implica em um ganho de conhecimento e de inteligência de processo, pois se passará a ter um registro das variáveis do bioprocessamento como temperaturas, iluminâncias, concentração de biomassa, umidade, dentre outras.

A necessidade de usar uma plataforma de hardware e software de baixa manutenção, que seja aberta, que seja acessível, não só em custo, mas em facilidade de uso, justifica a escolha da plataforma Arduino, para servir como base para a construção de equipamento capaz de monitorar o cultivo.

Para prover as informações de concentração de biomassa durante a operação do fotobiorreator é necessário haver um sensor de biomassa microalgal; dentre as possibilidades, há o uso de sensores ópticos, baseados na medição da intensidade da radiação luminosa passante através da suspensão de cultivo ou no espalhamento da radiação luminosa pela suspensão de microalgas.

É desejável também que o conjunto composto pelo hardware e software de acompanhamento do cultivo seja avaliado quanto a possibilidade de realizar a modelagem cinética durante o cultivo. Essa avaliação é necessária pelas características peculiares do hardware baseado em microcontrolador, em geral associado com tarefas mais simples, menos dispendiosas do ponto de vista computacional.

Por esses motivos, este trabalho foi baseado na concepção, desenvolvimento e teste de uma plataforma de acompanhamento de cultivo em fotobiorreator *outdoor raceway* baseada em microcontrolador Arduino.

3.1 Histórico

O grupo de pesquisadores do Laboratório de Engenharia Bioquímica (LEB) da Universidade Federal do Rio Grande estuda microalgas desde 1996, desenvolvendo trabalhos

sobre cultivo, produção e uso de biomassa. O primeiro projeto de graduação em Engenharia de Alimentos versando sobre o tema (ATALA et al.) ocorreu no ano de 1997.

Os trabalhos pioneiros envolveram a modelagem e o estudo das condições de cultivo de microalgas, em escala de bancada, como o trabalho de LINDE et al. (2000) que efetuaram a modelagem das condições de crescimento para a cianobactéria *Spirulina platensis*. KOZZA et al. (2001), também em escala de bancada, avaliaram diferentes fontes de nitrogênio e as respectivas respostas de crescimento para a *Spirulina platensis*.

Prosseguindo na evolução dos esforços, e fazendo a ampliação de escala, COLLA et al. (2003) acompanharam o crescimento da microalga *Spirulina platensis* em fotobiorreatores abertos tipo *raceway*, usando água suplementada com carbono, nitrogênio e alguns cofatores de crescimento suplementares.

Mais recentemente, houve a construção da estufa *outdoor* com planta piloto de produção de microalgas, usando fotobiorreatores do tipo *raceway*, no Campus Cidade da Universidade Federal do Rio Grande e a implantação da planta industrial de produção de microalgas, em Santa Vitória do Palmar, no sul do Rio Grande do Sul, próximo a Lagoa Mangueira (em regime de cooperação com a Fundação ZERI), dotada de 3 fotobiorreatores de 15 m³, com capacidade para produção de até 70 kg/mês de biomassa (MORAIS et al., 2009)

Já foram estudados substratos alternativos para o crescimento, configurações de fotobiorreatores e modos de cultivo, efeito de fatores como temperatura, iluminância, taxa de renovação e concentração de corte, de custos de produção, potencial antitumoral e hipocolesterolêmico, modelagem matemática do crescimento de *Spirulina*, produção de ácidos graxos, isolamento de uma cepa nativa do extremo sul do Brasil. Foram desenvolvidos alimento com adição de biomassa microalgal (MORAIS et al., 2006), e suplementos alimentares para praticantes de atividades físicas adicionados de *Spirulina* (CARVALHO, 2010). Foram estudadas a biofixação de gás carbônico oriundo da combustão (ROSA et al., 2011; RADMANN et al., 2011; MORAIS; COSTA, 2008b; RADMANN; COSTA, 2008; CAMERINI, 2008) e a seleção de microalgas para produção de bioetanol (MARGARITES, 2010).

As atividades de pesquisa e desenvolvimento se estenderam também, desde 2004, para o acompanhamento técnico do projeto, construção e implantação da biofixação de dióxido de carbono dos gases de chaminé da termoeletrica à carvão de Candiota (projeto em parceria técnica com a CGTEE e ELETROBRAS).

A pesquisa sobre microalgas no LEB também abordou a produção de biogás usando biomassa de microalgas (HENRARD, 2013), produção de lipídeos a partir da microalga *Chlorella* (FERREIRA, 2013), nanopartículas com peptídeos bioativos oriundos de microalgas (LISBOA, 2013), produção de carboidratos e bioetanol (MARGARITES, 2014), produção de biosurfactantes e nanoemulsões a partir de microalgas (CARVALHO, 2014) e a síntese extração e purificação de biopolímeros utilizando microalgas (MARTINS, 2014).

Dentro do contexto histórico das pesquisas do Laboratório de Engenharia Bioquímica da FURG, este trabalho se inseriu com o intuito de agregar elementos de automação e controle de processos ao extenso trabalho já existente, bem como contribuir com a iniciativa de desenvolver aplicações computacionais em microntroladores voltadas para o estudo de microalgas.

CAPÍTULO II

4 REVISÃO DA LITERATURA

A partir do intuito de abarcar o monitoramento do cultivo de microalgas e cianobactérias em fotobiorreator *raceway* é necessária uma revisão dos fatores associados não só com o bioprocesso em si, mas também com os aspectos que são relevantes para que seja possível monitorar a operação de fotobiorreatores em planta piloto, usando um dispositivo de hardware especificamente construído para esse fim, usando microcontroladores Arduino.

Serão abordadas características da biomassa microalgal, os biorreatores para produção de biomassa microalgal, a natureza dinâmica do cultivo de microalgas, os fundamentos da lógica *fuzzy*, a plataforma de hardware Arduino na construção de *dataloggers*, os sensores ópticos de concentração de biomassa, a otimização por enxame de partículas e os sensores por software.

4.1 Biomassa Microalgal

As cianobactérias são micro-organismos unicelulares procariotos fotossintéticos que têm funções importantes no sequestro global de carbono biológico, na produção de oxigênio e no ciclo do nitrogênio (PARAMAR et al., 2011). Vestígios fósseis de cianobactérias têm sido encontrados a partir de cerca de 3,5 bilhões de anos atrás, e as mesmas, muito provavelmente desempenharam papel fundamental na formação do oxigênio atmosférico, e na evolução dos cloroplastos de algas e plantas verdes (TAMAGNINI et al., 2007).

As aplicações da biomassa microalgal são diversas, tais como: alimento (BARROS,2010) , fonte de pigmentos, produção de ácidos graxos, de lipídios, de amido, na produção de bioenergia, no sequestro de CO₂ (AHSAN et al., 2008).

A biomassa proveniente do cultivo de cianobactérias pode ser usada diretamente como fonte de alimento humano (MORAIS et al., 2006) ou de matérias-primas para grupos de biomoléculas importantes, tais como: antioxidantes, pigmentos, e compostos bioativos.

Além do verdadeiro portfólio de compostos derivados de interesse, há a possibilidade de produção de bioenergia a partir da biomassa de microalgas, que pode ser convertida em biometano (biogás) sob digestão anaeróbia (ANDRADE et al. 2009).

O ciclo de Calvin leva à produção de carboidratos, proteínas, lipídios e ácidos graxos. Hidratos de carbono podem ser convertidos em bioetanol por fermentação. Lipídios podem ser convertidos em biodiesel (CHRISTENSON et al., 2011).

Estudos têm mostrado as aplicações bioenergéticas das microalgas na produção de biocombustíveis (CHISTI, 2007), e também na mitigação do acúmulo de gases geradores do efeito estufa (MORAIS & COSTA, 2007).

Segundo Morais e Costa (2011) as vantagens da utilização de microalgas na produção de biocombustíveis são fundamentadas na sua produção contínua, no ciclo de divisão celular simples, na biosíntese de compostos orgânicos através da fotossíntese, na tolerância às diferentes condições ambientais, na utilização de efluentes ou da água salobra, no uso da terra não utilizada para a agricultura.

O uso do gênero *Spirulina* data da civilização Asteca, onde a biomassa era coletada de lagos de águas alcalinas para consumo alimentício (AHSAN et al., 2008). O cultivo extensivo deste gênero é realizado devido as suas propriedades, como o teor proteico (60 a 70%) e o conteúdo de carboidratos de reserva, especificamente o amido das cianofíceas. Este amido modificado tem estrutura análoga a do amido, no entanto não possui zonas cristalinas tão regulares (MORAIS et al., 2009).

As características morfológicas típicas da *Spirulina* são a forma regularmente espiralada, com diâmetro da hélice de 20 a 45 μm , largura do filamento de 6 a 14 μm , grau de espiralação de 80 a 140 e comprimento total de 400 a 900 μm (WU et al., 2012).

4.2 Biorreatores para Produção de Biomassa Microalgal

O cultivo comercial de microalgas é feito, com apenas algumas exceções, em sistemas abertos (RICHMOND et al., 1999). A principal razão para isto é que grandes tanques abertos são mais fáceis e menos caros para construir e operar, e mais duráveis do que os grandes reatores fechados, sendo o projeto de construção essencialmente vinculado às condições locais e materiais disponíveis (BECKER, 1994 apud RICHMOND, 1999).

Apesar dos diferentes tipos de lagoas propostas, apenas três grandes projetos têm sido desenvolvidos e operados em uma escala relativamente grande: sistemas inclinados onde a mistura é conseguida através de bombeamento e do fluxo por gravidade; lagoas circulares

com agitação fornecida por um braço giratório; lagoas do tipo *raceway* construídas como um laço infinito, no qual o cultivo é movimentado por pás motorizadas. Somente os dois últimos, juntamente com lagoas naturais, são usados para a produção comercial de microalgas (RICHMOND, 1999).

Lagoas abertas são os sistemas mais antigos e mais simples para cultivo em massa de microalgas. Neste sistema, a lagoa rasa é geralmente de cerca de 30 cm de profundidade, e as algas são cultivadas em condições semelhantes aquelas do meio ambiente. Apesar dos biorreatores do tipo lagoa apresentarem um investimento fixo e um custo variável menor do que fotobiorreatores fechados apresentam desvantagens inerentes ao seu tipo de projeto construtivo. Na medida em que são sistemas em contato com a atmosfera apresentam uma significativa perda de água devido à evaporação (TREDICI, 2004; RICHMOND, 1999).

Assim, lagoas abertas para cultivo de microalgas não permitem usar o dióxido de carbono da forma mais eficiente, e a produção de biomassa é limitada (CHISTI, 2007).

A produtividade de biomassa microalgal também é limitada pela contaminação com microalgas de espécies indesejadas, bem como organismos predadores que se alimentam de algas (TREDICI, 2004). Além disso, condições de cultivo ideal são difíceis de manter em tanques abertos, e a recuperação da biomassa de tal cultivo diluído é cara (MOLINA GRIMA, 1999).

Fotobiorreatores fechados têm sido empregados para superar os problemas de contaminação e evaporação encontrados em lagoas abertas (MOLINA GRIMA, 1999). Estes sistemas são feitos de materiais transparentes e geralmente são colocados ao ar livre para usufruir da iluminação natural. Esses biorreatores devem dispor de uma grande razão entre área superficial e volume (TREDICI, 2004).

O processo de fotossíntese gera oxigênio; em um sistema aberto, este não é um problema, pois o oxigênio é simplesmente devolvido à atmosfera. No entanto, em fotobiorreatores fechados, os níveis de oxigênio irão aumentar até que comecem a inibir e envenenar as microalgas. Dessa forma o cultivo deve, periodicamente retornar a uma zona de degaseificação onde o meio de cultivo com as microalgas é aerado para remover o excesso de oxigênio (RICHMOND, 1999).

A composição do meio de cultivo é um fator fundamental na produção de biomassa com características desejadas, além de afetar a taxa de crescimento e o nível máximo da produção de biomassa (SÁNCHEZ, 2000).

A manipulação de condições de cultivo, através da presença ou ausência, redução ou acréscimo de determinados nutrientes, estimula a biossíntese de compostos que vão desde enzimas até fármacos (RICHMOND, 1999).

A principal espécie química é o íon bicarbonato que entra nas células por transporte ativo e, através da enzima anidrase carbônica, libera o dióxido de carbono, o qual é disponibilizado para uso no ciclo de Calvin, na via anabólica de produção de carboidratos, proteínas e lipídios (KAPLAN & REINHOLD, 1999).

Segundo Vonshak e Richmond (1988), um aspecto muito relevante a ser considerado no projeto de fotobiorreatores é o fenômeno da fotolimitação, que ocorre devido ao sombreamento e a competição pela luz que as células causam entre si, no interior do cultivo.

4.3 Natureza Dinâmica do Cultivo de Microalgas

A identificação da dinâmica dos sistemas a serem controlados é uma etapa fundamental na automação clássica de processos produtivos (SEBORG, et al. 2003). Essa se dá através da representação por meio de funções de transferência, expressões matemáticas no domínio complexo (s), que relacionam a variável de entrada, $X(s)$, com a variável de saída, $Y(s)$.

Essa representação é limitada, pois é inteiramente válida somente para sistemas dinâmicos descritos por equações diferenciais ordinárias (EDOs) lineares de coeficientes constantes, chamados de sistemas invariantes no tempo (OGATA, 2003).

Ocorre que a produção em grande escala de biomassa microalgal não é descrita perfeitamente como um sistema invariante no tempo, mesmo quando feita de forma contínua, pois podem ocorrer variações, difíceis de estimar e computar, como por exemplo a perda de até 25% da concentração celular de biomassa na etapa escura do ciclo de luz (CHRISTI, 2007). Acrescente-se a isso o fato de que biorreatores produtores de microalgas operam a maioria das vezes de forma descontínua, o que caracteriza o regime de operação transiente, o

que dificulta o projeto de sistemas de controle automático clássicos para esse tipo de equipamento (LUYBEN & LUYBEN, 1996).

Outra demanda importante é a forma de representação do crescimento microalgal ao longo do tempo, a cinética de crescimento, que tem um comportamento esperado descrito pela curva de crescimento, mas que apresenta variabilidade, em função dos distúrbios nas variáveis que influenciam o crescimento e até mesmo na evolução biológica da própria cultura.

Para a representação matemática da curva de crescimento, podem ser usadas as equações do modelo logístico de Verhulst (PELEG, 2007), ou o modelo do crescimento de micro-organismo sobre condições de restrição de substrato de Monod (SCHMIDEL, 2001), ou o modelo modificado de Gompertz (DASTIDAR,2006) ou representações baseadas na modelagem de Baranyi e colaboradores (1994).

Esta variedade de modelos advindos do campo da microbiologia preditiva, se justifica devido as particularidades associadas com cada tipo de micro-organismo, em função das condições globais de cultivo e de manutenção.

Velten (2009) classifica os modelos matemáticos em modelos fenomenológicos e modelos mecanísticos. Fenomenológicos são aqueles baseados somente em dados experimentais, e por levarem em consideração as informações de entrada e de saída do sistema, são também chamados modelos “black box”. Já os modelos mecanísticos recebem esse nome ao se fundamentarem no estudo dos mecanismos internos do sistema, a partir da disponibilidade de toda a informação sobre o mesmo, sendo assim chamados de modelos “white box”.

A maioria dos modelos usados em engenharia encontra-se num patamar entre os modelos mecanísticos puros e os “black box”, sendo chamados de semi-empíricos ou “gray box models”.

O uso de modelos estatísticos, dirigidos aos dados e baseados em regressão, é comum, sendo os modelos lineares, do tipo uma variável dependente e uma independente a forma mais comum e simplificada possível. Mesmo modelos multivariáveis lineares, ou modelos não lineares baseados em linearização são comuns (LORENA & SOUZA,2005).

Outra abordagem é o uso de ferramentas de inteligência artificial no desenvolvimento de modelos híbridos, que unem a inteligência artificial e o ramo do conhecimento em questão, com o intuito de criar modelos empíricos ou semi-empíricos (BAUGHMAN & LIU, 1996).

Vega et al. (2011) analisaram a influência da temperatura sobre o crescimento e composição da biomassa de *Spirulina maxima* construindo modelos empíricos baseados em redes neurais, que descreveram a taxa de crescimento celular em função da temperatura. Além disso, um modelo de rede neural generalizada foi construída, capaz de prever a concentração celular máxima e as características de composição da biomassa de *Spirulina maxima* (proteínas, carboidratos e níveis de γ -linolênico), na faixa de temperatura de cultivo (20 °C - 40°C).

Furlong et al. (2011) construíram modelos empíricos para a predição do crescimento da microalga *Synechococcus nidulans* fundamentados no uso de redes *neurofuzzy*, na implementação do tipo *Local Linear Model Tree (LOLIMOT)*, desenvolvida inicialmente por COLLETE (2007), através do uso do pacote matemático *open source* SCILAB.

4.4 Fundamentos da Lógica *Fuzzy* ou Nebulosa

A lógica nebulosa foi um dos desenvolvimentos decorrentes da teoria dos conjuntos *fuzzy* (ZADEH, 1965) e foi primariamente projetada para representar e trabalhar com o conhecimento que não pode ser expresso por medidas quantitativas. A principal ideia dos algoritmos baseados em lógica *fuzzy* é imitar o raciocínio humano para controlar processos de difícil modelagem, grande transitoriedade ou forte não linearidade (DUTRA PEREIRA, et al. 1999).

Os sistemas de inferência nebulosa modelam os aspectos qualitativos do conhecimento humano através de regras linguísticas “se-então” (ZADEH, 1965). É um ramo rigoroso da matemática, que dá origem aos números *fuzzy*, suas propriedades e operações (MUSSI et al., 2009).

Segundo Guerra, Oliveira e Sandri (1998) a lógica nebulosa faz parte do campo da ciência da computação chamado computação suave (ZADEH, 1994), englobando inclusive

outras metodologias como redes neurais e algoritmos genéticos; segundo os mesmos autores, a lógica nebulosa, quando utilizada em sistemas de controle, desempenha um importante papel no desenvolvimento de sistemas inteligentes para aplicações industriais e espaciais.

Os algoritmos clássicos de controle lidam com sistemas que tem um modelo matemático (mecânico ou empírico) precisamente definido (SEABORG, et al., 2010). No projeto de um controlador *fuzzy* não se necessita conhecer o modelo matemático do processo a ser controlado. Se o modelo existe, no entanto, pode ser usado para a simulação e para o teste da estratégia de controle. As principais vantagens dos controladores *fuzzy*, segundo GODJEVAC (1995) são: não necessitam de um modelo matemático do processo; permitem implementar o conhecimento e a experiência de um especialista humano usando regras linguísticas compreensíveis; - possibilitam controlar processos não-lineares.

São relatados diversos usos da lógica *fuzzy* em aplicações de engenharia das mais diversas, desde a década de 1970, tanto em aplicações de controle automático e modelagem, como sistema de controle de turbinas, controle do tratamento de águas, controle de temperatura, controle de pH, controle de reatores de polimerização, controle da destilação em batelada, (PROCYK e MAMDANI, 1979; YAGISHITA, ITOH e SUGENO, 1985; PENG e LIU, 1988; KARR, 1993; BIASIZZO et al., 1997; DUTRA PEREIRA, et al. 1999).

Em linhas gerais, sistemas *fuzzy* tratam as informações quantitativas de entrada, a partir das definições de um banco de dados interno composto pelas funções de pertinência aos conjuntos *fuzzy*, de forma a “fuzzificar” as entradas. A partir de então, o mecanismo de inferência, que vai determinar as saídas, pode gerar saídas numéricas, através de funções matemáticas, como o mecanismo de Takagi, Sugeno, Kang (TSK) ou saídas ainda no universo de discurso *fuzzy*, como o mecanismo de Mamdani (DUTRA PEREIRA, et al., 1999).

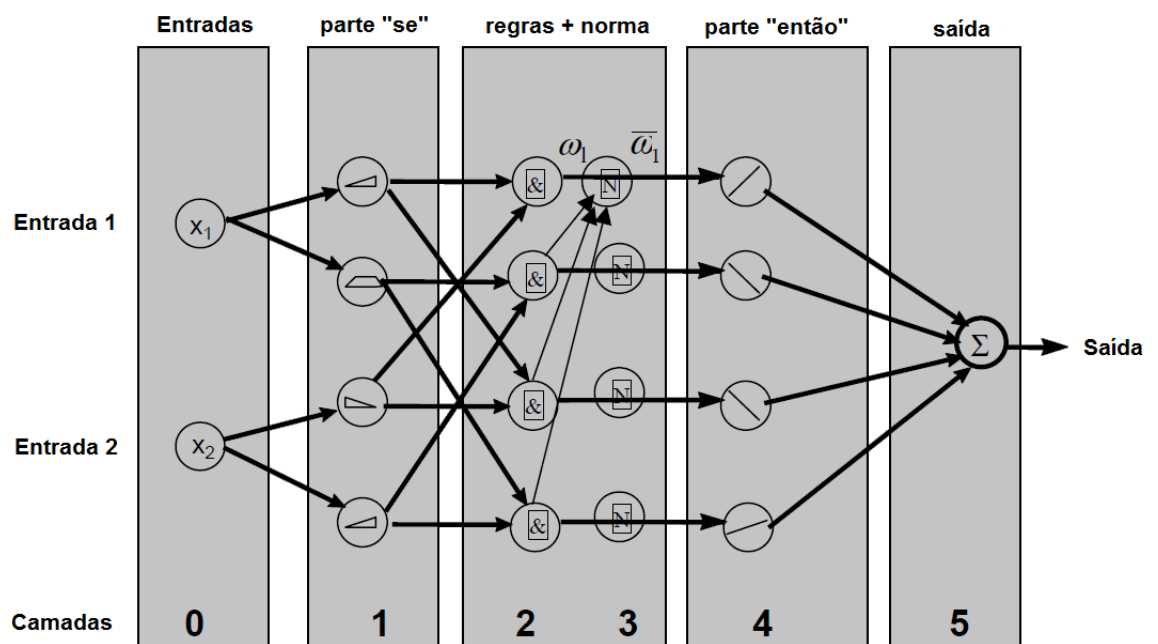
A vantagem do mecanismo de inferência TSK sobre o mecanismo de Mamdani é já gerar uma saída no universo de discurso quantitativo, e não em variáveis linguísticas, *fuzzy*, não necessitando da etapa de defuzzificação de variáveis.

Nos dois mecanismos, em no desenvolvimento de qualquer sistema *fuzzy*, há a necessidade de gerar a base de conhecimento compreendida pela definição das funções de pertinência *fuzzy*, pelo menos de entrada, e também de saída, no caso do mecanismo de Mamdani; há de se ter também a chamada base de regras, que constitui a forma como as variáveis linguísticas de entrada estão vinculadas com as variáveis linguísticas de saída.

O uso da lógica *fuzzy* e de *soft computing* em problemas de engenharia biológica e na agricultura já ocorre há mais de trinta anos (HUANG et al. ,2010), pois um conjunto de técnicas computacionais "inexatas" são propícias para modelar e analisar problemas complexos, oferecendo soluções com tolerância à imprecisão e incerteza, visto que a lógica *fuzzy* utiliza variáveis linguísticas multi-valor, não-numéricas para modelar o raciocínio humano aproximado (ZADEH,1975). A tolerância à imprecisão e à incerteza está vinculada a possibilidade de trabalhar na ausência de limites estritos (BAUGHMAN & LIU, 1996).

De acordo com Rashid & Ahmed (2012) o sistema de inferência adaptativo *neurofuzzy* (ANFIS) é um tipo de rede neural baseado no sistema de inferência *fuzzy* do tipo TSK - Takagi, Sugeno e Kang (1985), o qual integra tanto princípios de redes neurais como de lógica *fuzzy*, com potencial de agregar os benefícios de ambos simultaneamente – a capacidade de realizar o “raciocínio aproximado” e de simultaneamente armazenar conhecimento através do modelo conexionista, como mostra a Figura 2.

Figura 1 – Exemplo de Topologia de um Sistema ANFIS Genérico (adaptado de Bonissone, 2002).



O modelo TSK combina conjuntos *fuzzy* na parte antecedente da regra (a parte se) com funções “crisp” na saída: Se (x_1 é A) e (x_2 é B) então $y = f(x_1, x_2)$.

O sistema de inferência *fuzzy* corresponde a um conjunto de regras SE-ENTÃO que tem a capacidade de aproximar funções não lineares (GODJEVAC, 1995)

Os sistemas conexionistas baseados em redes neurais artificiais tem sido usados na automação de bioprocessos (GADKAR, 2005). Wali e colaboradores (2012) usaram a arquitetura de redes ANFIS com treinamento fundamentado em algoritmos genéticos para o controle online da temperatura no reator de produção de biodiesel usando micro-ondas, com o software Labview©, onde bons resultados experimentais mostraram que a arquitetura usada tem desempenhos semelhantes ao controle adaptativo tradicional baseado em modelos matemáticos.

Ao modelar a qualidade do efluente da digestão anaeróbia Erdirencelebi e Yalpir (2011) reafirmaram a capacidade de sistemas ANFIS em construir qualquer função não-linear, com múltiplas entradas e saídas, avaliando o desempenho do modelo desenvolvido por meio do coeficiente de correlação (R^2), do erro percentual médio (MPE) e o erro quadrático médio (RMSE).

4.5 A Plataforma de Hardware Arduino na Construção de *Dataloggers*

O microcontrolador Arduino é um hardware de controle *open source* o qual é projetado para ser facilmente integrado aos sensores e atuadores (www.arduino.cc). Apresenta como importantes característica a facilidade de programação mesmo por usuários com pouca experiência técnica na programação de hardware embarcado (LEUNG,2013).

A plataforma Arduino foi concebida em 2005 no Instituto Ivrea por Barragan, Banzi, Cuartielles, Mellis, Marino e Zambeti (GIBB, 2010), motivados pela necessidade de ensinar em pouco tempo eletrônica prática para estudantes de design do Instituto Ivrea (BANZI, 2012).

A plataforma passou a ser usada amplamente por *hackers*, amadores, *makers* e cientistas em projetos de tamanho pequeno e médio, o que resultou em inúmeras outras placas eletrônicas de extensão, para agregar funcionalidades (*shields*) às placas Arduino (VICENTE, 2012).

Recentemente a plataforma Arduino passou a ser usada para os mais diferentes fins, por pesquisadores dos mais diversos campos, tais como biomedicina, automação, robótica, engenharia (CELA et al.,2013;ORTEGA-ZAMORANO et al.,2013;KORNUTA et al., 2013).

Há exemplos de aplicação da plataforma Arduino em *datalogging* (ELMEHRAZ et al., 2013; PURVIS et al. 2013, BORKAR, 2012, HLOUPIS et al., 2012) em automação de bioprocessos (FALK, 2011). As aplicações citadas focam em monitoramento, automação, na construção de dispositivos standalone, reais, não no uso do Arduino em computação científica. Um dos motivos para isso é que enquanto constituinte de um sistema embarcado, o microcontrolador não apresenta uma interface de entrada/saída pronta. A mesma deve ser implementada especificamente para cada aplicação.

A utilização de microcontroladores de pouco poder computacional é restrita na simulação digital e na modelagem cinética de bioprocessos, quando comparado com o uso de outras plataformas de hardware computacional, pelo fato da necessidade de implementar em linguagem C ou C++ a aplicação computacional. Como o campo da computação numérica digital faz extensivo uso do poder computacional, não só por exigências dos problemas cujos dados são tratados, modelados, e simulados, mas também devido à necessidade de executar os pacotes computacionais e seus *toolboxes*, como o Matlab, conforme Khaouane e colaboradores (2013) que fizeram a modelagem do processo fermentativo de produção do antibiótico pleuromutilina com redes neurais, através do uso do *toolbox* de redes neurais disponível na ferramenta Matlab® , com o intuito de acelerar o desenvolvimento das aplicações computacionais.

Outro aspecto a ser considerado é o condizente com o consumo de energia durante o uso do hardware. De acordo com Göddeke e colaboradores (2013) o uso de soluções de computação baseadas em processadores de arquitetura ARM é mais eficiente energeticamente do que o uso, para a solução do mesmo problema, da plataforma x86.

Por outro lado o uso do Arduino como ferramenta de ensino de Engenharia, tem sido aplicado para estudantes das séries iniciais (BUTTERFIELD, 2013), para a realização de atividades práticas de aquisição de dados e automação de pequenos sistemas.

Sob o ponto de vista do poder computacional, da flexibilidade dos sistemas computacionais multi-propósito, e da interface com usuário, os sistemas computacionais x86 são melhores do que os microcontroladores, para a simulação. A grande vantagem do

microcontrolador é o baixíssimo custo, sua robustez, e pequena necessidade de componentes eletrônicos adicionais para uso. O chip ATMEGA 168 de baixo custo (US\$ 5,0), por exemplo, opera na faixa de -55°C até +125°C e necessita de uma fonte de tensão de 5V, um cristal oscilador de 16MHz e alguns capacitores (www.arduino.cc).

Ocorre que na construção de dispositivos de campo, que monitorem ou acompanhem bioprocessos, em planta piloto ou real grandeza, a possibilidade de contar com ferramentas de computação que permitam a construção de sistema embarcado é vantajosa. Assim, além de contar com hardware de monitoramento que registra informações e as grava ou transmite para pós-processamento, o processamento desses dados pode ser feito localmente no microcontrolador e fornecer os parâmetros para outras aplicações, como co-validação, decisão sobre estratégias de operação, conciliação de informações de balanço material, controle, dentre outras possíveis.

Com a disponibilidade da plataforma Arduino de hardware livre (ACOSTA & HIPPEL, 2009; BANZI, 2011), capaz de propiciar um rápido aprendizado por usuários sem conhecimento específico de eletrônica (MONK, 2011), começou a ocorrer o desenvolvimento de aplicações em ciência e tecnologia com base nas placas dotadas de microcontroladores, inicialmente em 8 bits, e após em 32 bits (BANZI,2011).

Há aplicações usando Arduino em robótica, em impressoras 3D, em CNC, em monitoramento atmosférico, no monitoramento de águas naturais, em agricultura, no cultivo de microalgas, em gadgets, em ensino, na arte, em automação (www.arduino.cc).

A disseminação da plataforma Arduino esta vinculado ao novo paradigma do “Open Hardware”. Open Hardware é o hardware eletrônico projetado e oferecido da mesma maneira que um software livre. As placas Arduinos estão sob a licença Creative Commons CC-SA-BY. Essa licença permite: compartilhar - copiar e redistribuir o material em qualquer suporte ou formato; adaptar - remixar, transformar, e criar a partir do material para qualquer fim, mesmo que comercial. É importante observar que no caso do usuário remixar, transformar, ou criar a partir do material, ele tem de distribuir as suas contribuições sob a mesma licença que o original (<http://creativecommons.org>).

Dessa maneira, com o projeto das placas Arduino inteiramente disponível, surgiram as placas compatíveis, disseminadas globalmente. Em meados de 2014 há mais de um milhão de

placas Arduino oficiais comercializadas, e provavelmente alguns outros milhões de placas compatíveis também comercializadas, de acordo com Massimo Banzi (2013).

Ocorre que o uso dessa plataforma tem limitações, como a memória e a velocidade de processamento, que devem ser levadas em conta no projeto das aplicações. Esses fatores foram levados em conta pela própria comunidade, visto que há um verdadeiro “ecossistema” de hardwares Arduino diferentes, desde as placas microcontroladas com menos recursos, Arduino Mini, Arduino Lilypad, Arduino Nano, até as placas com mais recursos como os Arduinos Due, Tre, e Arduino Galileo, parceria com a Intel.

A construção de dispositivos para realizar medições com Arduino é crescente, vide Gertz & Justo (2012) que construíram condutivímetro de baixo custo, usando a plataforma Arduino, para monitorar a qualidade de águas naturais.

4.6 Sensores Ópticos de Concentração de Biomassa Microalgal

De acordo com Beuermann et al. (2012) o monitoramento de processos microbiológicos usando sensores ópticos tem crescido de importância devido ao baixo custo dos sensores fotométricos, quando comparados com os espectrômetros, também devido à característica de serem compactos, facilmente calibráveis, robustos e passíveis de adaptação à geometria dos biorreatores.

O monitoramento em tempo real de cultivos de micro-organismos disponibiliza informações em fase com o estado do processo, possibilitando detecções rápidas de desvios da normalidade e, conseqüentemente, a reação imediata ao desvio (VOJINOVIC et al., 2006).

No entanto, a instalação no lugar de trabalho significa que o sensor deve ser capaz de manter sua integridade estrutural sob as condições de processo (TREVISAN & POPPI, 2006).

Para implementar uma estratégia de automação no cultivo de microalgas é necessária a determinação automática de concentração de biomassa. A utilização de instrumentação baseada em medidas de densidade óptica, como uma medida turbidimétrica, é a metodologia de determinação não invasiva mais comum de biomassa em bioprocessos (MEIRELES et al, 2008), sendo usada como método de referência online e *offline* (MAROSE et al., 1999).

Os problemas associados com medidas turbidimétrica on-line de biomassa em reatores são a não-linearidade, a interferência de substâncias coloridas, de partículas e de bolhas, além do crescimento de organismos nas superfícies de emissores e receptores (COX et al. 1989).

Castelijm e colaboradores (2000) usaram o conceito de sensor baseado na medição dupla da turbidez, combinando o espalhamento e a medição da transmissão da radiação no meio de cultura com o intuito de reduzir instabilidades provocadas pela variabilidade das propriedades ópticas e pela geometria das partículas, além das dificuldades operacionais dos sensores ópticos de biomassa.

Cogne e colaboradores (2001) relataram problemas na medição da concentração de biomassa in-situ em cultivos do gênero *Spirulina* devido a deposição nos instrumentos e as variações dos pigmentos do micro-organismo, o que limita a faixa de utilização do sensor óptico no mesmo.

Marose e colaboradores (1999) elencaram as seguintes vantagens para o uso de sistemas de sensores ópticos para o monitoramento de bioprocessos: resposta rápida, versatilidade, medidas in vivo não-invasivas, não destrutivas e sem perturbar o mecanismo celular, ausência de perturbação no processo de cultivo, possibilidade de medição de parâmetros intracelulares e extracelulares, medidas in situ sem necessidade de amostragem, sem diluição e/ou adição de reagentes, sem preparo de analito, fácil manutenção, sem interação com campos elétricos ou magnéticos, custos relativamente baixos, sistemas de sensores resistentes a ambientes hostis e inapropriados.

A interpretação de grandes quantidades de dados é frequentemente um problema, no entanto o uso de modelos computacionais e quimiométricos, tais como PCA, análise de componentes principais, ou redes neurais artificiais permitem a extração de importante informação dessa quantidade de dados (MAROSE et al., 1999)

A dispersão e a absorção da radiação luminosa, tem uma dependência não-linear com a concentração em massa seca, que inviabiliza o uso de curvas padrão lineares na calibração (SHULER & KARGI, 2001).

A modelagem matemática dos fenômenos de difração, absorção, extinção e espalhamento da luz é caracterizada pela interação da radiação com a matéria, e recai na

solução das equações de Maxwell, conforme a relação de tamanho de partícula com o comprimento de onda, tipo de radiação e geometria das partículas (ZANGWILL, 2012)

O tratamento matemático dado por Mie (1908), por exemplo, fica prejudicado para o problema caracterizado pela interação da radiação luminosa com a suspensão de cianobactéria *Spirulina* sp. LEB 18. Além de não levar em conta a dispersão de tamanho das células, a solução de Mie (1908) envolve o uso de uma geometria uniforme para cada partícula, em formato de esfera perfeita. A presença de pigmentos com forte absorção (GRIFFITHS, 2011) em algumas bandas do espectro visível também prejudica o uso desse modelo.

Esses modelos matemáticos para geometrias uniformes que descrevem a interação entre a radiação luminosa e suspensões de partículas, apesar de não serem apropriados para as suspensões de micro-organismos de morfologia não-uniforme, trazem a informação qualitativa de que a dispersão da luz é maior na direção direta (no sentido emissor-receptor) do que no reverso e que quanto maior a partícula mais luz é espalhada na direção direta (ZANGWILL, 2012).

4.7 A Otimização por Enxame de Partículas

De acordo com o dicionário Merriam-Webster (2014) a otimização é o campo da matemática aplicada cujos princípios e métodos são usados para solucionar problemas quantitativos na física, biologia, engenharia e economia. Questões de maximização ou minimização de funções de diversas áreas podem ser usadas com as mesmas ferramentas matemáticas.

A solução ótima, $\mathbf{z} \in \mathfrak{R}^n$, atende ao melhor critério de avaliação e satisfaz a Equação (minimização):

$$\mathbf{z}^n, \mathbf{y}^n \in \mathfrak{R}^n : f(\mathbf{z}) \leq f(\mathbf{y}) \quad (1)$$

A busca pela solução ótima pode ser feita iterativamente seguindo o gradiente, ∇f .

Quando a função não é diferenciável ou seu gradiente é desconhecido uma opção é usar a otimização heurística, sendo que a mesma faz uso de métodos de solução que englobam conhecimento empírico sobre o problema. Dentre os métodos heurísticos, aqueles advindos do campo da computação natural, como o método da otimização por colônia de formigas ou a otimização por enxame de partículas, se apresentam como alternativas.

O algoritmo original de otimização por enxame de partículas (*Particle Swarm Optimization* - PSO), foi desenvolvido por Kennedy e Eberhart (1995), e teve origem nas pesquisas sobre “vida artificial” e no modelo de cultura adaptativa de Kennedy.

A otimização pelo enxame de partículas (Kennedy e Eberhart, 1995; Kennedy e Eberhart, 2001) faz uso de um paradigma computacional fundamentado no fenômeno da inteligência coletiva exibida por enxames de insetos, cardumes de peixes e bando de pássaros, que tem sido aplicada na solução de diversos problemas de otimização dos mais variados campos de pesquisa (Eberhart e Shi, 2001).

Um “enxame” é um coletivo aparentemente desorganizado de indivíduos que tendem a se aglomerar enquanto que cada indivíduo parece estar se movendo em uma direção aleatória. O “enxame” poder ter seu comportamento descrito por certos comportamentos sociais.

A otimização por enxame de partículas é conceitualmente simples, fácil de implementar, computacionalmente eficiente, efetiva para uma grande variedade de problemas, expansível para a busca multidimensional.

A população de indivíduos, possíveis soluções do problema de otimização, é inicializada a partir da atribuição de posições e velocidades randômicas, soluções essas que irão “transitar” através do espaço multidimensional de soluções.

O algoritmo registra a melhor posição de cada partícula, quanto a um índice de avaliação (ou *fitness*), “pbest”; já a população inteira apresenta uma melhor solução candidata global, “gbest”; A cada iteração, cada partícula estocasticamente acelera, entre “pbest” e “gbest”.

O pseudo-código fundamental do algoritmo PSO, segundo Eberhart e Kennedy (1995) é:

1 – inicializar a população no espaço de resposta;

2 – avaliar quanto à solução as partículas individualmente;

3 – modificar as velocidades com base na melhor posição global e da melhor posição individual segundo as Equações de atualização 2 e 3;

$$v_{i+1,d} = w \cdot v_{i,d} + c_1 \cdot \text{rand}().(\text{pbest}_{id} - x_{id}) + c_2 \cdot \text{rand}().(\text{gbest}_d - x_{id}) \quad (2)$$

$$x_{i+1,d} = x_{i,d} + v_{i,d} \quad (3)$$

4 – caso a condição de término for alcançada, encerrar;

5 – voltar ao passo 2.

onde:

$v_{i+1,d}$ é a velocidade na próxima iteração na dimensão d , $v_{i,d}$ é a velocidade i na dimensão d , w é a inércia, c_1 e c_2 são constantes positivas entre 0 e 2, $\text{rand}()$ é a função geradora do fator randômico, $pbest_{i,d}$ é a melhor posição da partícula, $gbest_d$ melhor posição da população, $x_{i,d}$ é a posição da partícula i , $x_{i+1,d}$ e $x_{i,d}$ são, respectivamente, a posição x na próxima iteração e a posição atual, na dimensão d .

O peso da componente cognitiva, c_1 , representa a ponderação da experiência individual da partícula de onde a solução se encontra. Já o peso da componente social, c_2 , representa a ponderação da experiência do conjunto de partículas, enxame, de onde a solução se encontra.

A otimização por enxame de partículas tem como vantagens:

- insensibilidade à mudança de escala das variáveis;
- implementação simples;
- adaptabilidade à computação paralela;
- não requer cálculo de derivadas;
- poucos parâmetros a serem definidos pelo usuário.

Por outro lado, como desvantagem, o algoritmo PSO apresenta um desempenho rápido para localizar a bacia de atração das boas soluções, mas lento no ajuste fino da solução (como nos algoritmos genéticos).

4.8 Sensores por Software

Modelos matemáticos de processos podem ser utilizados em conjunto com sensores para estimar variáveis de processo, que não podem ser medidas diretamente. Esta abordagem – baseada na teoria de controle automático - é amplamente utilizada, inclusive em engenharia de bioprocessos; tais estimadores baseados em computação foram denominados “sensores de software” (*soft sensors*), também chamados sensores virtuais, refletindo o fato de que eles podem substituir ou apoiar sensores clássicos, baseados em hardware (HAVLIK et al., 2013). Outros autores, como Pitasse e Fileti (2001) adotam a nomenclatura de sensores virtuais.

Esses sensores fazem uso, de alguma forma, da estimação de estado, cujo objetivo é estimar as variáveis de estado de um sistema a partir de determinadas medições online/inline (LU, 1996). Em geral, as aplicações destas técnicas envolvem predição, filtragem e suavização de informações (GELB, 1974).

Embora os sistemas de sensores existentes para o cultivo de microalgas possam fornecer medições on-line para os parâmetros físicos de interesse em cultivos, sistemas de monitoramento de importantes variáveis químicas e biológicas não são tão bem desenvolvidos, o que ocorre devido ao elevado grau de complexidade das células e do seu ambiente químico (HAVLIK et al., 2013).

Algumas das funções dos sensores de software são: a redução do ruído e do tempo de resposta; o primeiro é conseguido pela filtragem do sinal digital e o último por meio de um modelo de processo matemático ou uma correlação.

Vários tipos de estimadores foram estabelecidos: aqueles dirigidos aos dados - estimadores empíricos representados por correlações ou por redes neurais artificiais,, enquanto estimadores baseados em modelo de processo são construídos por meios das equações de conservação - balanços de massa e energia - acoplados à observadores de estado adaptativos, por exemplo, como o filtro de Kalman (HAVLIK et al., 2013).

Redes neurais podem ser usadas como funções flexíveis de regressão não linear, pois propiciam um grande número de parâmetros de ajuste que podem ser usados para aproximar qualquer função (VELTEN, 2009).

Conforme Haykin (1994) o aprendizado em RNAs é um processo no qual os parâmetros livres da rede neural são adaptados através de um processo contínuo de

estimulação pelo ambiente no qual a rede está incorporada. O tipo de aprendizado é determinado pela maneira pela qual a mudança dos parâmetros ocorre.

Dentre os tipos de treinamento, o treinamento pela correção do erro faz uso de métodos de otimização com o intuito de minimizar uma função objetivo, definida como o sinal de erro, a diferença entre a resposta desejada, $d_k(n)$, e a resposta atual, $y_k(n)$.

Em resumo, o desenvolvimento dos sensores por software utiliza fundamentos e as aplicações do campo da inteligência artificial, para intuir sobre o processo em estudo, usando a informação disponível, de forma indireta, quando o uso exclusivo dos modelos mecânicos não é a melhor opção.

CAPÍTULO III

5 DESENVOLVIMENTO DO TRABALHO

O trabalho foi dividido em quatro artigos científicos:

ARTIGO I: Sistema Autônomo de Monitoramento do Cultivo de Biomassa Microalgal: Desenvolvimento e Teste.

ARTIGO II: Efeito da Concentração de *Spirulina* sp. LEB 18 no Espalhamento da Radiação Luminosa Monocromática

ARTIGO III: Sensor Óptico Combinado Definido por Software para Determinação da Concentração de *Spirulina* sp. LEB 18

ARTIGO IV: Microcontroladores Arduino na Modelagem da Cinética de Bioprocessos Autotróficos: desenvolvimento e desempenho.

Com o intuito de documentar as montagens dos protótipos e as aplicações computacionais desenvolvidas ao longo do trabalho, os códigos das mesmas estão assim apresentados, bem como os registros fotográficos:

- o Apêndice A apresenta um conjunto de fotografias que ilustram o transcorrer do trabalho;
- o software desenvolvido para o microcontrolador Arduino e usado nos protótipos construídos e testados, conforme relatado no Artigo I, encontra-se no Apêndice B;
- para realizar a ativação do sistema de emissores laser, registro de leituras dos sensores ópticos, conforme será descrito no Artigo II, foi construído e implementado um software, para o Arduino Due, apresentado no Apêndice C;
- no Apêndice D está o software para treinamento do sistema ANFIS com o algoritmo PSO, descrito no Artigo III;
- o Apêndice E trás o software de modelagem cinética através da equação de Verhulst, testado nas placas Arduino Nano, Mega e Due, conforme o Artigo IV;

ARTIGO I: Sistema Autônomo de Monitoramento do Cultivo de Biomassa Microalgal: Desenvolvimento e Teste.

RESUMO

O cultivo de microalgas é uma alternativa para a biofixação de dióxido de carbono, bem como fonte de biomassa, matéria-prima para a indústria alimentícia. A monitoração automática no cultivo é útil no acompanhamento das variáveis fundamentais à produção de biomassa em larga escala, tais como luminosidade, temperatura e concentração de biomassa no cultivo, bem como na tomada de decisão sobre a operação do cultivo. Este trabalho teve como objetivo principal prototipar um equipamento autônomo de acompanhamento da cinética de cultivo em tempo real, denominado MARK (*Microalgal Analyzer of Real Time Kynetics*). O sistema de aquisição de dados é composto por conjunto de sensores ópticos de biomassa desenvolvidos e testados ao longo do trabalho. O sistema armazenador de dados concebido é gerenciado por um microcontrolador Arduino Nano, e dotado de módulo de tempo real (RTC) e gravador de dados em micro SD. Foram desenvolvidos durante o trabalho quatro protótipos, que operaram em reais condições de cultivo em estufa, entre os meses de outubro de 2012 à janeiro de 2013. Os sistemas desenvolvidos foram resistentes às condições de temperatura, bem como de umidade no interior da estufa, propiciando operação contínua durante a realização dos experimentos. Foram acompanhados a luminosidade no interior da estufa, dentro do cultivo, a temperatura na estufa e dentro do cultivo, a concentração de biomassa através de sensores ópticos e pH. Houve correlação estatística entre o sinal do sistema óptico desenvolvido e a concentração de biomassa somente para concentrações de biomassa abaixo de 0,6 g/L. A razão entre a iluminância dentro e fora do cultivo caiu em uma razão que sugere uma relação exponencial, ao longo do cultivo, entre 47% e 42% da iluminância disponível fora do cultivo. O uso da plataforma de hardware livre Arduino acelerou o ciclo de desenvolvimento do sistema desenvolvido, bem como se mostrou suficientemente robusta, exata e com precisão suficiente no sistema de aquisição de dados para o acompanhamento da cinética de cultivo.

Palavras-chave: cultivo de microalgas, iluminância, microcontrolador Arduino, sensor

Microalgae Biomass Production Autonomous Monitoring System: Development and Test

ABSTRACT

Microalgae culture is an possible path to carbon dioxide biofixation, while producing the biomass itself, a high value ingredient to food products. Monitoring fundamental variables in the culture is an important step towards efficient industrial scale process. Such variables may be luminosity, biomass concentration and temperature and may enable in automated actuation within the system, in order to increase the process productivity. This work objective was to build a culture data logger prototype, named MARK (Microalgal Analyzer of Real Time Kinetics). The data acquisition system is composed by a array of optical biomass sensors developed and tested throughout this work. The system hardware is conceived and built around a Arduino Nano microcontroller. The microcontroller is aided by a Real Time Clock (RTC) and micro SD dock. Four prototypes were manufactured and all were subjected to real culture conditions in greenhouse between the months of October 2012 and January 2013. The systems were developed in order to operate in the high temperature and humidity inside the greenhouse, enabling continuous operation during the experiments duration. The monitored variables were luminosity and temperature, both inside the greenhouse and in the culture, greenhouse humidity, biomass concentration through optical sensor and culture pH. A positive correlation between the biomass concentration and the measured optical signal was only achieved for concentration less than 0.6 g.L^{-1} . The difference between the greenhouse ambient luminosity and inside the culture decreased exponentially from 47% to 42% throughout the culture. The utilization of the Arduino free platform accelerated the prototype development and presented itself as a fairly robust solution. The Arduino architecture also presented sufficient numeric precision in order to monitor the growth kinetics.

Keywords: Arduino microcontroller, datalogging, microalge, sensors

1 INTRODUÇÃO

O acompanhamento do processo de produção em larga escala de biomassa microalgal normalmente é feito através de amostragens e posteriores análises *offline*, acarretando um atraso no conhecimento das condições, especificamente da concentração da biomassa.

Conforme há o investimento na intensificação dos processos de biofixação autotrófica de dióxido de carbono, através da pesquisa e utilização de diferentes configurações de reatores, cresce a necessidade de adotar técnicas de instrumentação e monitoramento de processos.

Este trabalho teve por objetivos desenvolver um sistema de acompanhamento de cultivo de microalgas, montar o hardware do sistema de monitoramento, desenvolver o software de monitoramento, registro e tratamento das informações obtidas dos cultivos e inferir à respeito do cultivo de *Spirulina* sp. LEB 18, em cultivos em reatores do tipo *raceway* de 600L, em condições de operação em planta piloto, sob as reais condições de operação de estufa de cultivo.

Surgiram como aspectos a serem acompanhados durante o transcorrer do trabalho a viabilidade de uso da plataforma de hardware livre, não só quanto às suas características funcionais de hardware e memória, associadas com o microprocessador usado na mesma, mas também quanto às condições ambientais de operação na estufa, sujeito às condições reais de iluminação, temperatura e umidade.

O projeto e a operação de fotobiorreatores das mais diversas configurações, dedicados à produção de biomassa microalgal, tem sido amplamente estudadas em laboratório e escala piloto (RICHMOND & TREDICI, 2007). O interesse pela produção de biomassa de *Spirulina* é vinculado ao seu potencial uso como fonte de proteínas e pigmentos, dentre outros (AHSAN et al., 2008)

O acompanhamento das variáveis de processo no cultivo de cianobactérias e microalgas é feito, em geral, *offline* (MAROSE et al., 1999), o que é comum para muitos processos biotecnológicos, em especial aqueles realizados em condições operacionais reais de produção (VOJINOVIĆ et al., 2006). Como investigação em escala laboratorial há registro da aplicação de técnicas de monitoramento e inclusive automação em malha fechada,

especialmente em aplicações de fotobioreatores para produção de oxigênio e biomassa, visando a indústria espacial (STAVREV, 2011).

Com a disponibilidade da plataforma Arduino de hardware livre, também chamado de *open hardware* em um conceito semelhante ao do *open software* (ACOSTA & HIPPEL, 2009; BANZI, 2009), começou a ocorrer o desenvolvimento de aplicações em ciência e tecnologia das placas dotadas de microcontroladores, inicialmente em 8 bits, e após em 32 bits (BANZI, 2009). Há aplicações de Arduino em robótica, em monitoramento atmosférico, em agricultura, no cultivo de microalgas, em gadgets, em ensino, na arte, em automação (www.arduino.cc). Fisher e Gould (2012) afirmam que o uso de microcontroladores é uma alternativa viável para a construção de instrumentação analítica e de registro de informações viável e de baixo custo.

Na realização do acompanhamento da concentração de biomassa *inline*, a utilização de instrumentação baseada em densidade ótica como uma medida turbidimétrica é a metodologia de determinação não invasiva mais comum em bioprocessos (MEIRELES et al., 2008), sendo usada como método de referência *online* e *offline* (MAROSE et al., 1999).

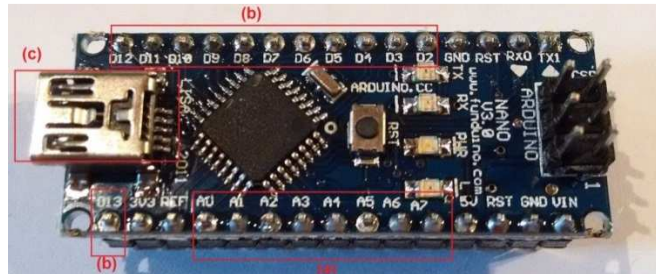
2. MATERIAIS E MÉTODOS

O cultivo de *Spirulina* sp. LEB 18 ocorreu em escala piloto em reator aberto do tipo *raceway* coberto por uma estufa de filme translúcido com proteção UV. Os tanques utilizados tem aproximadamente 600 L de volume útil total.

O meio utilizado foi o Zarrouk (ZARROUK, 1966), com 50% da concentração original da fonte de carbono com o intuito de reduzir a perda química de dióxido de carbono para a atmosfera (ANDRADE et al., 2008). O processo de cultivo foi realizado em regime semi-contínuo, sendo os cortes realizados quando a concentração atingida era a desejada. A concentração celular da microalga *Spirulina* sp. LEB 18 foi determinada diariamente por análise de massa seca (ZITELLI et al., 2000).

Para compor a unidade de aquisição, tratamento e registro das informações de cultivo, foi usado o microcontrolador ATmega328, em uma implementação compatível com a placa Arduino Nano (Funduino, China), mostrado na Figura 1, que foi programado usando a interface padrão de desenvolvimento de software Arduino (www.arduino.cc).

Figura 1 - Placa do Arduino Nano v3.0. (a) Portas Analógicas. (b) Portas Digitais.(c) Porta micro USB.



A Figura 1 mostra a placa Arduino Nano, dotada de microcontrolador ATmega328, porta micro USB, operando em 5V (nível lógico), com voltagem recomendada de (7-12V), limites de voltagem de alimentação de 6 a 20V, 14 portas digitais (D2 até D13 mais RX,TX usadas para comunicação serial) das quais 6 portas são capazes de emular saída analógica com modulação por largura de pulso (PWM), 8 entradas analógicas, com corrente elétrica limitada em 40 mA por pino, 32KB de memória dos quais 2 KB usadas pelo *bootloader*, 2 KB de memória SRAM, 1KB de EEPROM, 16 MHz de velocidade de clock, e dimensões de 19 mm x 43 mm.

Com o intuito de adquirir dados sobre a radiação luminosa fora e dentro dos cultivos, bem como serem usados como elementos primários de detecção nos sensores ópticos de biomassa testados, foram usados nos protótipos desenvolvidos sensores de luz do tipo resistor dependente da luz - LDR (Brasil), fotocélulas de silício policristalino (SunWell, Taiwan) e sensor TAOS (Texas Advanced Optoelectronic Solutions Inc., USA).

As temperaturas dos cultivos foram acompanhadas a partir do uso do sensor eletrônico LM35, devidamente encapsulado (Lab. de Garagem, Brasil) e dentro da estufa pelo uso do sensor digital DHT11 (Lab. de Garagem, Brasil)

Foi desenvolvido sensor óptico de biomassa microalgal (SO 1.0) com base na medição da radiação monocromática passante no meio de cultivo. O sistema óptico foi acoplado à base fixa para mantê-lo com a distância entre a porção emissora e receptora constante, esta distância foi de 10 mm, O emissor de radiação monocromática usado foi do tipo laser vermelho de AlGaInP em 660 nm (Wuhan Besram Technology Inc., China).

A operação dos sensores foi comandada pelos protótipos baseados no microcontrolador Arduino, denominados MARKs (*Microalgae Analyzer of Realtime Kinetics*), equipamentos concebidos e construídos com o intuito de acompanhar a cinética de crescimento dos micro-organismos fotossintéticos, com intervalo de aquisição de informações de 5 em 5 min, funcionando como *dataloggers* (BERGMANN, WALLACE & CALIA, 2010).

Foi acompanhada a luminosidade na estufa e no interior do cultivo, com sensores do tipo célula de silício policristalino (no cultivo) e com sensor do tipo LDR - resistor dependente da luz (na estufa).

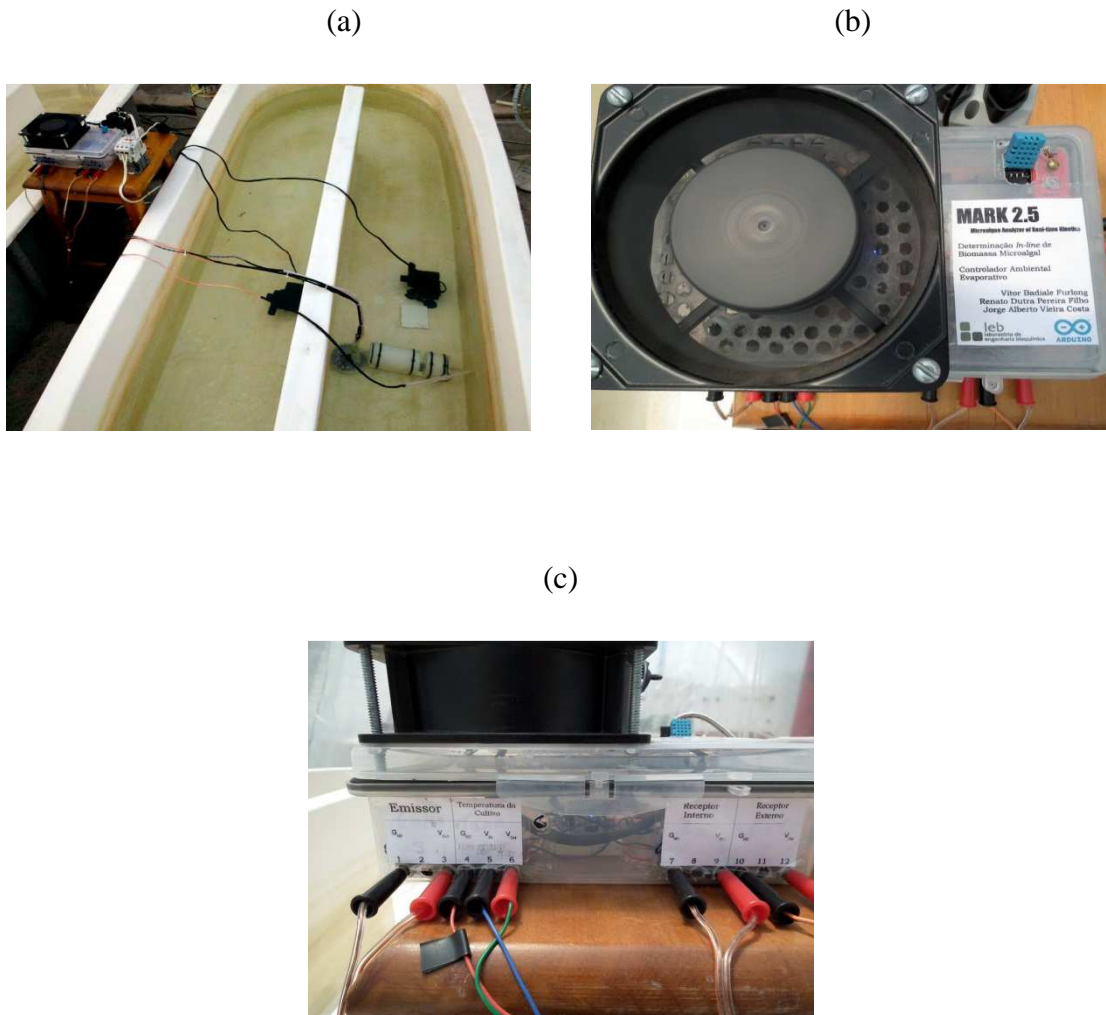
O sistema MARK 1.0 foi controlado por Arduino Nano, onde foi acoplado o SO 1.0, posicionado no interior do reator de cultivo *raceway*; o hardware microcontrolado coordenava a ativação do emissor laser e adquiria as intensidades luminosas no receptor SO 1.0 no interior do cultivo e no ambiente da estufa através de LDR posicionado na parte superior do equipamento. Os dados foram arquivados em um cartão de memória SD para posterior análise dos dados, com tempo de amostragem de 5 min.

A evolução da concepção de projeto do MARK 1.0 levou à substituição do mesmo pelo sistema MARK 2.0. Este ainda possuía os equipamentos presentes no MARK 1.0, e foi adicionado ainda o sensor DHT11 de umidade e temperatura ambiente - sensor capacitivo capaz de gerar um sinal digital passível de ser adquirido pelo Arduino. O protótipo recebeu ainda um sensor de temperatura para acompanhar a temperatura dentro do cultivo. Este sensor foi um LM35 - circuito integrado que gera uma voltagem proporcional à temperatura a qual está submetido (TEXAS INSTRUMENTS, 2000). O protótipo desenvolvido ainda recebeu um cooler para arrefecimento das porções computacionais e foi rebatizado de MARK 2.5.

3. RESULTADOS E DISCUSSÃO

A montagem experimental realizada no desenvolvimento dos experimentos de cultivo de *Spirulina* sp. LEB 18, bem como o protótipo do MARK 2.5 são apresentados na Figura 2.

Figura 2 - (a) Arranjo do Sistema de Monitoramento no *Raceway*; (b) Vista Geral do Protótipo de Monitoramento; (c) Ligações do *MARK* ao Emissor e Sensores



A presença do *cooler* no protótipo foi fundamental para propiciar o funcionamento devido às altas temperaturas no interior da caixa do protótipo, devido as temperaturas durante o uso do equipamento, conforme mostrado na Figura 3.

O conjunto de informações brutas obtidas pelo SO 1.0 dentro do cultivo em função do tempo, armazenadas pelo protótipo MARK 2.5, são mostradas na Figura 4.

Figura 3 – Registro de Temperaturas ao Longo do Cultivo de *Spirulina* na Estufa Outdoor pelo MARK 2.5

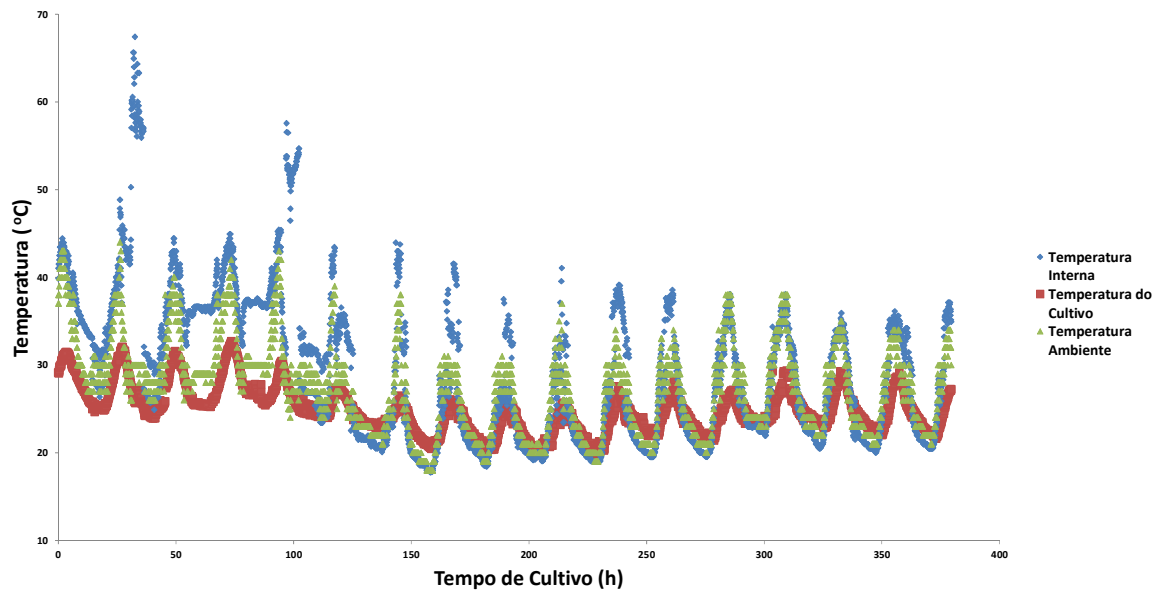
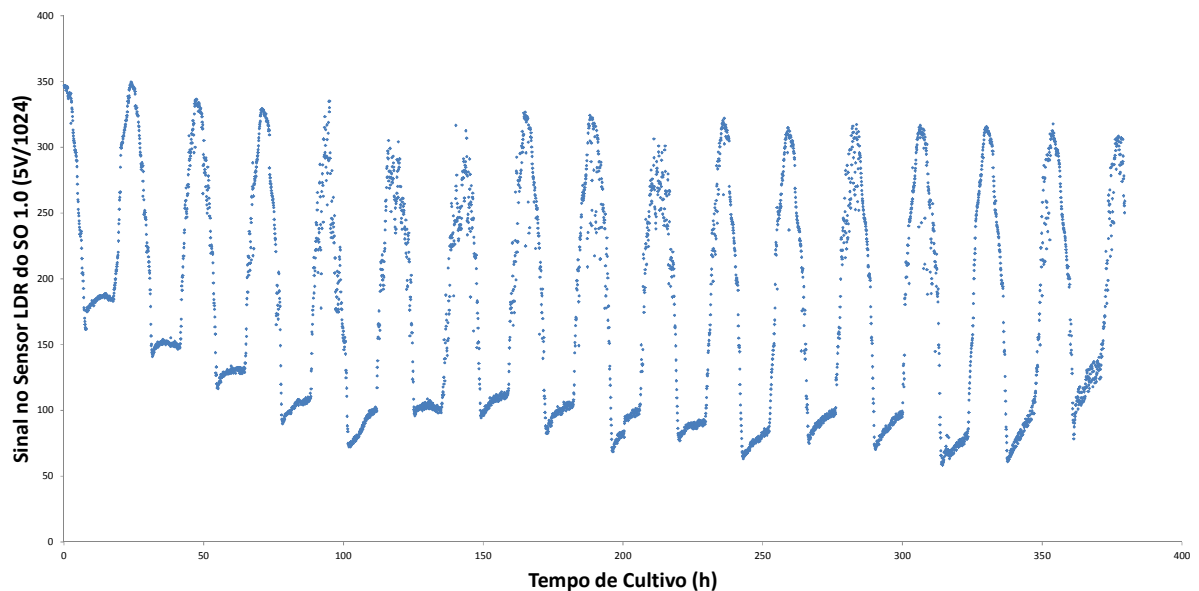


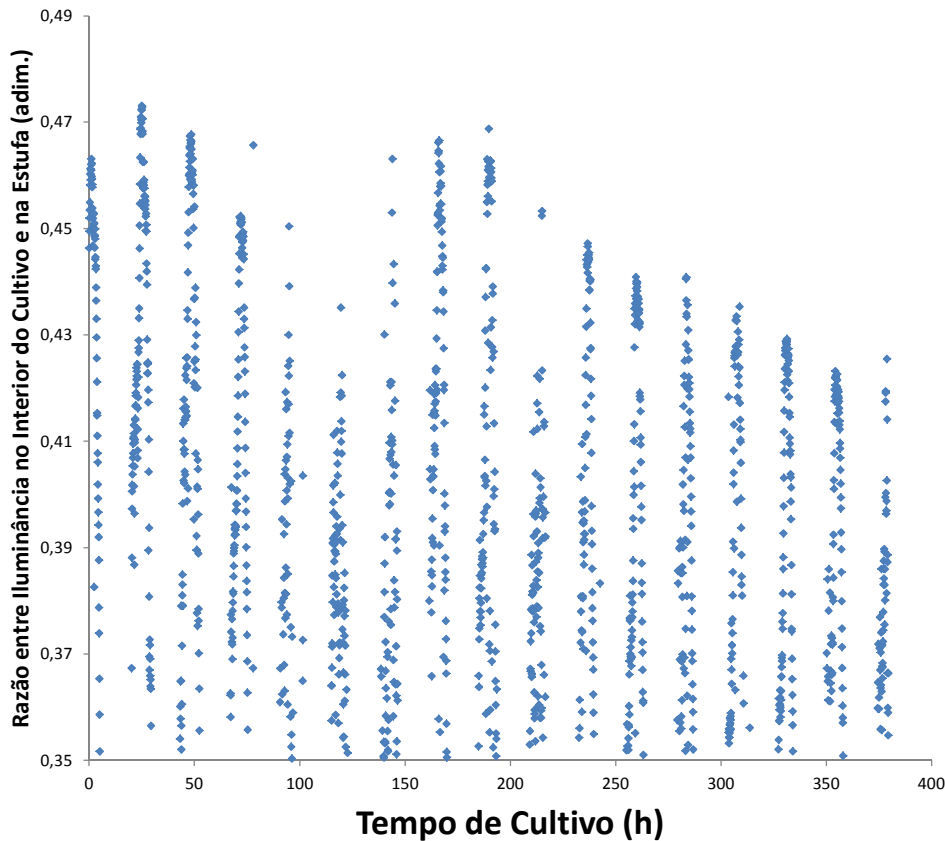
Figura 4 - Conjunto de medições realizadas de Iluminância no interior da estufa (azul) e no interior do cultivo de *Spirulina* sp. LEB 18 (vermelho).



A hipótese de que o aumento da concentração celular de biomassa de *Spirulina* sp. LEB 18 mantém relação com a parcela de radiação luminosa dissipada pelo meio de cultivo e

células pode ser avaliada, relacionando a evolução temporal da razão entre as duas medidas, conforme a Figura 4.

Figura 5 - Evolução da Razão entre a Iluminância no Interior do Cultivo de *Spirulina* sp. LEB 18 e no Interior da Estufa.

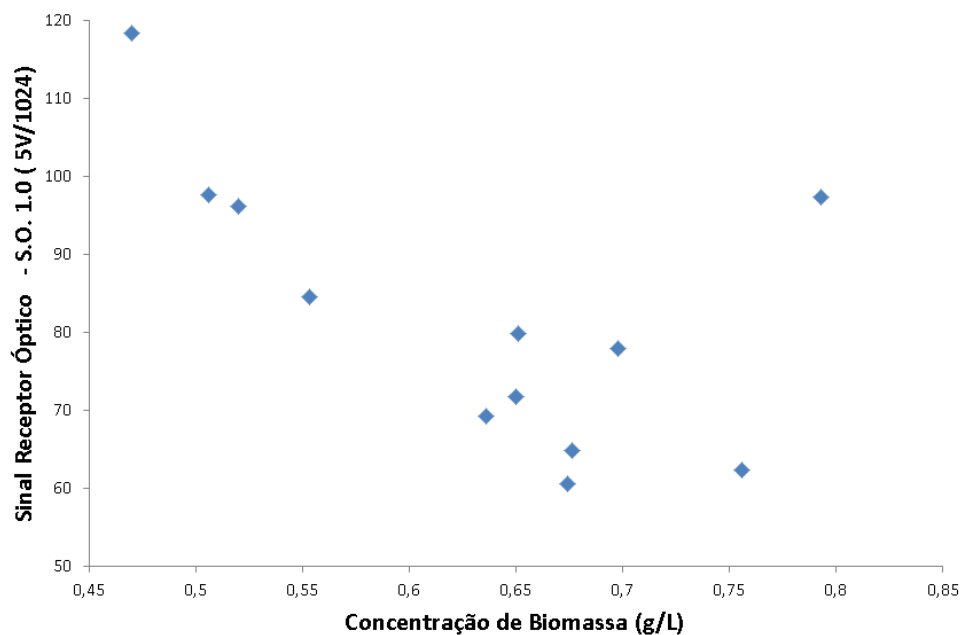


Conforme se pode observar a razão descrita na Figura 5 apresenta um comportamento inicial, em média, decrescente ao longo do tempo, que pode ser associado tanto com a dispersão da radiação pelo aumento da concentração da biomassa microalgal no cultivo, como pelo aumento de compostos coloridos, pigmentos, que podem absorver maior parcela da radiação luminosa disponível no cultivo, ao longo do tempo. Com o transcorrer do cultivo houve a interferência dos metabólitos produzidos ao longo do cultivo na razão entre as iluminâncias.

Não houve correlação estatística entre a concentração mássica de biomassa e o sinal no sensor óptico, durante o período diurno, pelo fato do mesmo ser exposto e a radiação luminosa solar alcançá-lo. Já durante o período noturno, detectado pela ausência de sinal no

sensor de luminosidade na estufa, houve baixa correlação entre o sinal no receptor óptico e a concentração diária de biomassa, sendo observável correlação somente para concentrações de biomassa inferiores a 0,6 g/L, conforme mostra a Figura 6.

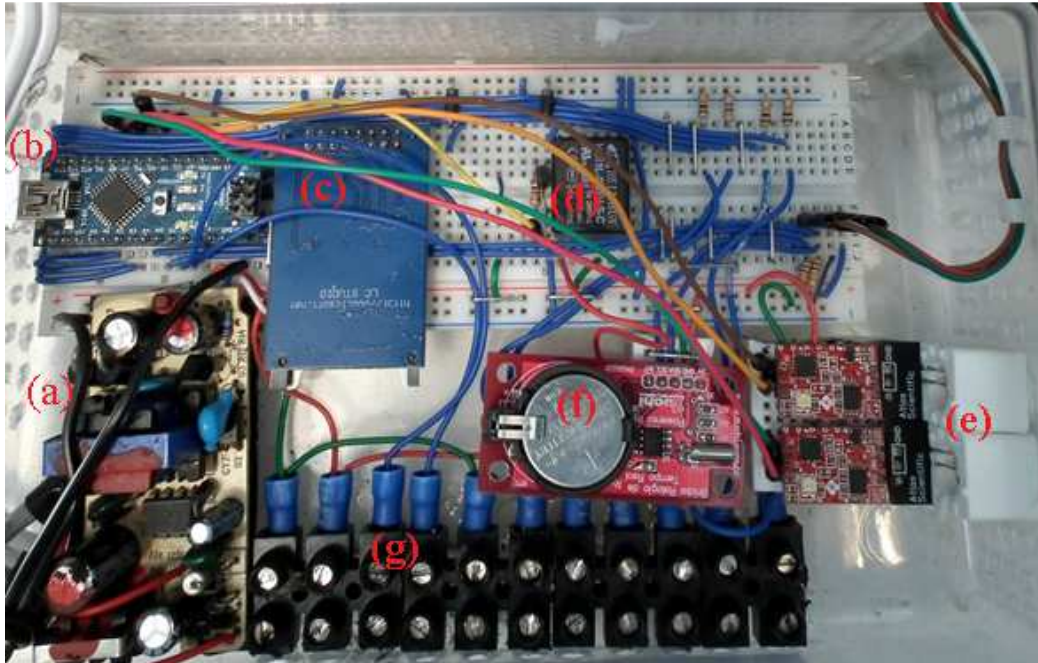
Figura 6 - Sinal do Receptor Óptico de Biomassa no Início do Período Noturno e Concentração de Biomassa.



Griffiths (2011) relata a forte interferência nos sensores ópticos de biomassa baseados em medida de radiação passante, não só pelos pigmentos, e demais materiais liberados no meio de cultivo, mas também pelos restos de células, morfologia, e estado dos vacúolos gasosos.

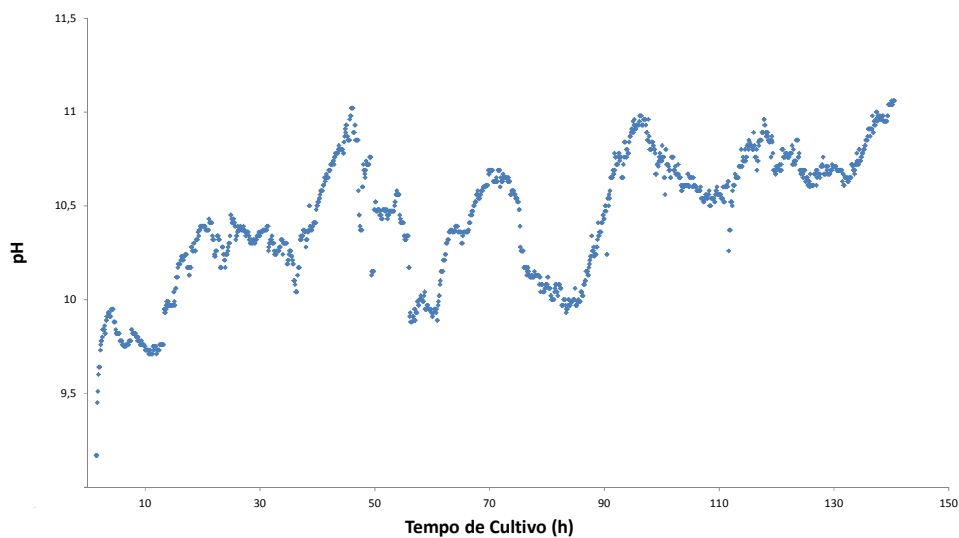
A Figura 7 mostra o interior da montagem do protótipo MARK 3.0, com o detalhe do uso do relógio de tempo real (RTC) e dos “Shields” de aquisição do sinal de pH, à direita na Figura 7. Um exemplo do registro de pH ao longo do tempo, para experimento com MARK 3.0 em *raceway* encontra-se na Figura 8.

Figura 7 – Organização Interna dos Componentes do Protótipo MARK 3.0 – (a) fonte de 6V; (b) Arduino Nano; (c) Leitor de Cartão SD; (d) Relé; (e) *Shield* de pH; (f) Relógio de Tempo Real (RTC); (g) Conexões



Conforme mostra a Figura 8, o pH durante o cultivo de *Spirulina*, mesmo com a redução de 50% da fonte de carbono (bicarbonato de sódio) é alto, e varia acompanhando o ciclo de iluminação, conforme varia o estado fisiológico da cultura.

Figura 8 - Registro pelo MARK 3.0 do pH ao Longo de Cultivo *Outdoor* de *Spirulina*



4 CONCLUSÕES

Foi desenvolvido um sistema de acompanhamento autônomo de cultivo de microalgas, foram montados protótipos de hardware para o sistema de monitoramento, e efetuado o registro das variáveis medidas ao longo dos cultivos de *Spirulina* sp. LEB 18 em fotobiorreator *raceway*, ao longo de corridas de cultivo de mais de 370h de duração, sob temperaturas de 18°C até 65°C.

As medidas dos sensores ópticos usando a medida da intensidade da radiação passante pela suspensão de *Spirulina* sp LEB 18 mostraram baixa correlação com a concentração. Esse resultado é indicativo da necessidade de avaliação da possibilidade de uso de outro fenômeno físico, que vincule outra propriedade óptica e a concentração da suspensão de microalga, como por exemplo o espalhamento da radiação.

A razão entre a iluminância no interior do cultivo e no interior da estufa caiu de 47% até 42%, ao longo do cultivo, mas de forma não linear, tanto acompanhando o ritmo da iluminação natural solar, mas indicando que os pigmentos e outros biocompostos produzidos no cultivo da microalga *Spirulina* interferem com a medição.

A evolução contínua da arquitetura dos protótipos, chegou até ao protótipo MARK 3.0, cujo desenvolvimento, como sistema autônomo de acompanhamento do cultivo, permitiu o registro de informações como temperatura, umidade, pH e luminosidade especialmente no interior do cultivo, permitindo melhores tomadas de decisão sobre o momento de efetuar o corte do cultivo, na operação semi-contínua de fotobiorreator *raceway*.

5 REFERÊNCIAS

- ACOSTA, R.; HIPPEL, E.; Open Source Hardware, Master's Thesis, MIT, 2009.
- AHSAN, M.; HABIB, M.; PARVIN, M.; HUNTINGTON, T.C.; HASAN, M. R.; A Review on Culture, Production and Use of *Spirulina* as Food for Humans and Feeds for Domestic Animals and Fish; Food and Agriculture Organization of the United Nations, FAO, Rome, 2008.
- ANDRADE, M. da R.; CAMERINI, F. V.; COSTA, J. A. V. Chemical carbon losses and growth kinetics in *Spirulina* cultures. *Química Nova*, v. 31, n. 8, p. 2031–2034, jan. 2008.
- BANZI, M.; Getting Started with Arduino, O'Reilly Media, 2009.

BERGMANN, N.W.; WALLACE, M.; CALIA, E.; Low cost prototyping system for sensor networks, 6th International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Brisbane, pp. 19-24, 2010.

FISHER, D. K.; GOULD, P.J.; Open-Source Hardware Is a Low-Cost Alternative for Scientific Instrumentation and Research, *Modern Instrumentation*, vol. 1, p.8-20, 2012.

GRIFFITHS, M. J. *et al.* Interference by pigment in the estimation of microalgal biomass concentration by optical density. *Journal of microbiological methods*, v. 85, n. 2, p. 119–123, 2011.

MAROSE, S.; LINDEMANN, C.; ULBER, R.; SHEPER, T.; Optical sensor systems for bioprocess monitoring. *Trends in Biotechnology*, v. 17, n. 1, p. 30–34, 1999.

MEIRELES, L. A. *et al.* On-line control of light intensity in a microalgal bioreactor using a novel automatic system. *Enzyme and Microbial Technology*, v. 42, n. 7, p. 554–559, 5 jun. 2008.

MONK, S.; *Programming Arduino: Getting Started with Sketches*, Tab Books, 2011.

RICHMOND & TREDICCI; *Handbook of Microalgal Culture: Biotechnology and Applied Phycology*, 2006.

STAVREV, V.; Dual-phase flow micro-gravity photobioreactor – Principles and an experimental module design, 41st International Conference on Environmental Systems, Portland, Oregon, 2011.

VOJINOVIĆ, V.; CABRAL, J. M. S.; FONSECA, L. P. Real-time bioprocess monitoring: Part I: In situ sensors. *Sensors and Actuators B: Chemical*, v. 114, n. 2, p. 1083–1091, Abril. 2006.

ZARROUK, C. Contribution à l'étude d'une Cyanophycée: influence de divers facteurs physiques et chimiques sur la croissance et la photosynthèse de *Spirulina maxima*. 1966. Tese (Ph.D) - Université Des Paris, Paris, 1966.

ARTIGO II: Efeito da Concentração de *Spirulina* sp. LEB 18 no Espalhamento da Radiação Luminosa Monocromática

RESUMO

O cultivo da cianobactéria *Spirulina* permite obter uma fonte de proteínas, pigmentos e outros biocompostos de interesse na indústria de alimentos. No acompanhamento *inline* da cinética de cultivo são fundamentais sensores de concentração de biomassa. Dentre esses sensores, os baseados em fenômenos ópticos apresentam-se como uma possibilidade a ser avaliada, especialmente pelo seu uso fora da região de validade da lei de Lambert-Beer. Este trabalho teve por objetivo avaliar a dispersão da radiação monocromática em suspensões da cianobactéria *Spirulina* sp. LEB 18, em condições controladas de temperatura e luminosidade sob escoamento com fluxo constante. Foi mensurada a intensidade da radiação espalhada detectada em função do ângulo formado entre emissor e receptor, da concentração de biomassa microalgal e do comprimento de onda usado. Durante a realização dos experimentos foi usado como suporte um dispositivo *in-situ* construído em plástico ABS através da técnica de impressão 3D, dotado de emissores lasers em 530 nm e 660 nm, receptores usando sensor TAOS TSL235R, sendo o sistema de aquisição de dados comandado por microcontrolador Arduino Due (32 bits). As medições de espalhamento da radiação monocromática em 530 nm pela suspensão de *Spirulina* que tiveram maior sensibilidade, na faixa entre 0,2 g/L e 1,4 g/L, foram aquelas com um ângulo entre emissor e receptor de 90°, sendo 9,1% em média maiores do que com o ângulo de 45°. A determinação da existência de correlação estatística ($R^2 > 0,95$) entre o sinal detectado pelo sensor TAOS TSL235R e a concentração de biomassa de *Spirulina*, durante o espalhamento da radiação de 530 nm, especificamente para os ângulos entre emissor e receptor de 45°, 90° e 135° reafirma a possibilidade de desenvolvimento de sensores ópticos de concentração de biomassa microalgal baseados em espalhamento da radiação para uso *inline*, sem necessidade de diluição da suspensão da *Spirulina*.

Palavras-chave: biomassa, determinação da concentração, sensor óptico

Effect of *Spirulina* sp. LEB 18 Biomass Concentration on Monochromatic Light Scattering

ABSTRACT

Spirulina culture enables the obtainment of a high value protein source, pigments and other interesting compounds to the food industry. In order to obtain such compounds in large quantities, it is necessary to monitor their production in the culture, thus, generating the necessity of keeping track with cellular growth. Optical sensors are a possible solution. However, further studies are necessary in order to generate a robust system, specially when working in concentration ranges outside the Lambert-Beer validity region. This work objective was to evaluate the dispersion of monochromatic radiation in different biomass concentrations of *Spirulina* sp. LEB 18, under controlled temperature, luminosity and flow conditions. The scattered radiation, from various emitters, was measured in different angles in relation with the original path direction under increasing biomass concentrations. An *in-situ* device was utilized in order to place the optical instruments in their correct place. The structure was first design with Computer Aided Design and extruded with 3D printing technique. Laser emitters with 530nm and 660nm radiations were mounted in the support to act as radiation emitters. Several sensor TAOS TSL235R were added different angles in order to measure dispersed radiation. The system was controlled by a microcontroller Arduino Due (32 bits). The measurements of scattering of monochromatic radiation at 530 nm by *Spirulina* suspension which had a higher sensitivity in the range between 0.2 g/L and 1.4 g/L were those with an angle of 90° between emitter and receiver, being 9.1% on average in the that the angle of 45°. The determination of statistical correlation ($R^2 > 0.95$) between the signal measured by the 45°, 90° and 135° receptors when the emitter of 530nm was utilized reassures the possibility of applying scattered light optical sensors in order to monitor inline biomass concentration without the dilution necessity.

Keywords: Biomass, Inline Determination, Optical Sensor

1 INTRODUÇÃO

A automação da operação em fotobiorreatores de grande escala permite a racionalização do uso de recursos sociais, econômicos e ambientais. O acompanhamento *inline* da concentração é uma maneira de monitorar a produção de biomassa microalgal em fotobiorreatores, que possibilita a automação.

O uso de medidas ópticas *inline* apresenta a vantagem operacional da não necessidade de diluição de amostras, das quais se deseja mensurar a concentração. Devem ser avaliados e identificados os parâmetros de construção e operação, além dos desvios decorrentes da operação em amplas faixas de concentração. Compreendem esses parâmetros: as faixas de concentração de biomassa onde os sensores ópticos podem operar, a distância entre os mesmos, o comprimento de onda da radiação monocromática usada e o ângulo formado entre emissor de radiação e receptor.

A dispersão da radiação monocromática em suspensões da cianobactéria *Spirulina* sp. LEB18 foi avaliada, em condições controladas de temperatura e luminosidade sob agitação. Foi mensurada a intensidade da radiação detectada em função do ângulo formado entre emissor e receptor, da concentração de biomassa microalgal e de dois comprimentos de onda usados para a radiação monocromática incidente.

De acordo com Beuermann et al. (2012) o monitoramento de processos microbiológicos usando sensores ópticos tem crescido de importância devido ao baixo preço dos sensores fotométricos, quando comparados com os espectrômetros. O monitoramento em tempo real de cultivos de micro-organismos disponibiliza informações em fase com o estado do processo, possibilitando detecções rápidas de desvios da normalidade e, consequentemente, a reação imediata ao desvio (VOJINOVIC et al, 2006).

No entanto, a instalação no lugar de trabalho significa que o sensor deve ser capaz de manter sua integridade estrutural sob as condições de processo (TREVISAN & POPPI, 2006).

Para implementar uma estratégia de automação no cultivo de microalgas é necessária a determinação automática de concentração de biomassa. A utilização de instrumentação baseada em densidade ótica como uma medida turbidimétrica é a metodologia de determinação não invasiva mais comum de biomassa em bioprocessos

(MEIRELES et al., 2008), sendo usada como método de referência *online* e *offline* (MAROSE et al., 1999).

Os problemas associados com medidas turbidimétricas on-line de biomassa em reatores são a não-linearidade, a interferência de substâncias coloridas, de partículas e de bolhas, além do crescimento de organismos nas superfícies de emissores e receptores (COX et al. 1989).

Castelijm e colaboradores (2000) usaram o conceito de sensor combinado, ou seja, baseado na medição dupla da turbidez, combinando o espalhamento e a medição da transmissão da radiação no meio de cultura com o intuito de reduzir instabilidades provocadas pela variabilidade das propriedades ópticas e pela geometria das partículas, além das dificuldades operacionais dos sensores ópticos de biomassa.

Marose e colaboradores (1999) elencaram as seguintes vantagens para o uso de sistemas de sensores ópticos para o monitoramento de bioprocessos: resposta rápida, versatilidade, medidas *in vivo* não-invasivas e não destrutivas, possibilidade de medição de parâmetros intracelulares e extracelulares, medidas *in situ* sem necessidade de amostragem, sem diluição e/ou adição de reagentes, sem preparo de analito, sem interação com campos elétricos ou magnéticos, e custos relativamente baixos.

Cogne e colaboradores (2001) relatam problemas na medição da concentração de biomassa *in situ* em cultivos do gênero *Spirulina* devido a deposição nos instrumentos e as variações dos pigmentos do micro-organismo, o que limita a faixa de utilização do sensor óptico no mesmo.

A interpretação de grandes quantidades de dados é frequentemente um problema, no entanto o uso de modelos computacionais e quimiométricos, por exemplo a Análise de Componentes Principais (PCA), que permitem a extração de importante informação dessa quantidade de dados (MAROSE et al., 1999)

A dispersão e a absorção da radiação luminosa, tem uma dependência não-linear com a concentração em massa seca, que inviabiliza o uso de curvas padrão lineares na calibração (SHULER & KARGI, 2001).

Os fenômenos de difração, absorção, extinção e espalhamento da luz são caracterizados pela interação da radiação com a matéria, e a modelagem matemática desses

fenômenos recai na solução das equações de Maxwell, conforme a relação de tamanho de partícula com o comprimento de onda, tipo de radiação e geometria das partículas (ZANGWILL, 2012)

O tratamento matemático dado por Mie (1908), por exemplo, fica prejudicado para o problema caracterizado pela interação da radiação luminosa com a suspensão de cianobactéria *Spirulina* sp. LEB 18. Além de não levar em conta a dispersão de tamanho das células, a solução de Mie (1908) envolve o uso de uma geometria uniforme para cada partícula, em formato de esfera perfeita. A presença de pigmentos com forte absorção (GRIFFITHS, 2011) em algumas bandas do espectro visível também prejudica o uso desse modelo.

Esses modelos matemáticos, para geometrias uniformes, que descrevem a interação entre a radiação luminosa e suspensões de partículas, apesar de não serem apropriados para as suspensões de micro-organismos de morfologia não-uniforme, trazem a informação qualitativa de que a dispersão da luz é maior na direção direta (no sentido emissor-receptor) do que no reverso e que quanto maior a partícula mais luz é espalhada na direção direta (ZANGWILL, 2012).

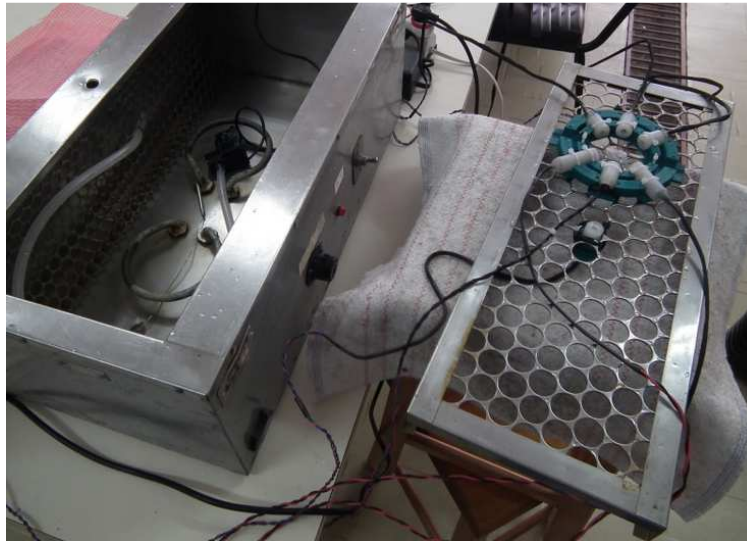
2. MATERIAIS E MÉTODOS

A biomassa de *Spirulina* sp LEB 18 foi produzida a partir de cultivo em escala semi-piloto em reatores abertos do tipo *raceway* cobertos por uma estufa de filme translúcido com proteção UV. Os tanques utilizados tem aproximadamente 600 L de volume útil.

O meio utilizado foi o Zarrouk (ZARROUK, 1966), com 50% da concentração original da fonte de carbono com o intuito de reduzir a perda química de dióxido de carbono para a atmosfera (ANDRADE et al., 2008). O processo de cultivo foi realizado em regime semi-contínuo, onde os cortes foram realizados quando a concentração atingida estava de acordo com a necessária para o processo de calibração. Neste ponto uma porção do volume era retirada e destinada ao processo de calibração e o volume do reator era restaurado ao volume inicial com o meio Zarrouk.

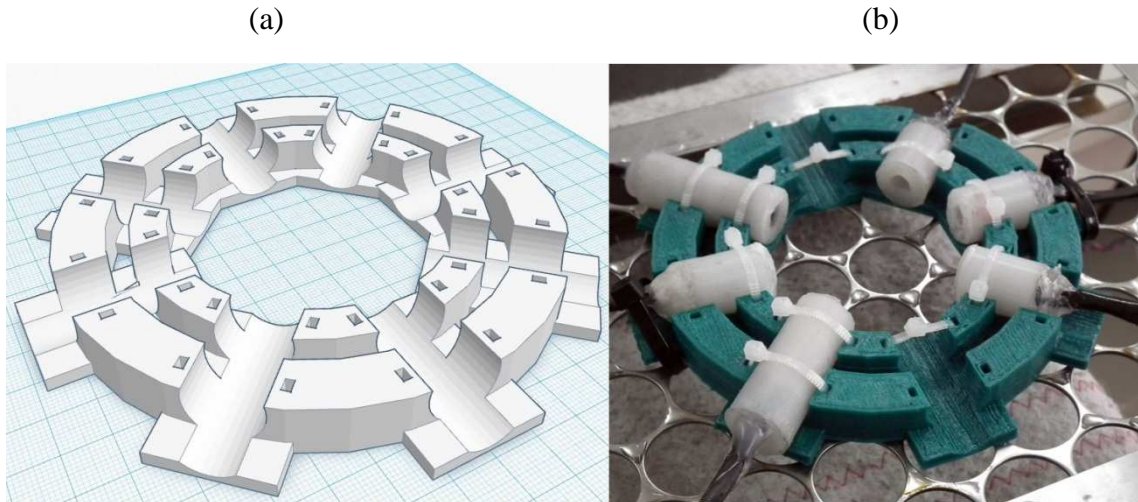
Para a realização dos experimentos ópticos foi adaptada uma câmara termostatzada, à 25°C, de 18 L de volume útil, para simular o espaço de um biorreator, no qual ficava colocado o suporte para os emissores e receptores ópticos, como mostra a Figura 1. Com o intuito de evitar a sedimentação das células, parte do conteúdo da suspensão era continuamente bombeado em ciclo fechado.

Figura 1 - Equipamento Usado para Realizar as Medidas de Espalhamento da Luz pela Suspensão de Microalga *Spirulina*



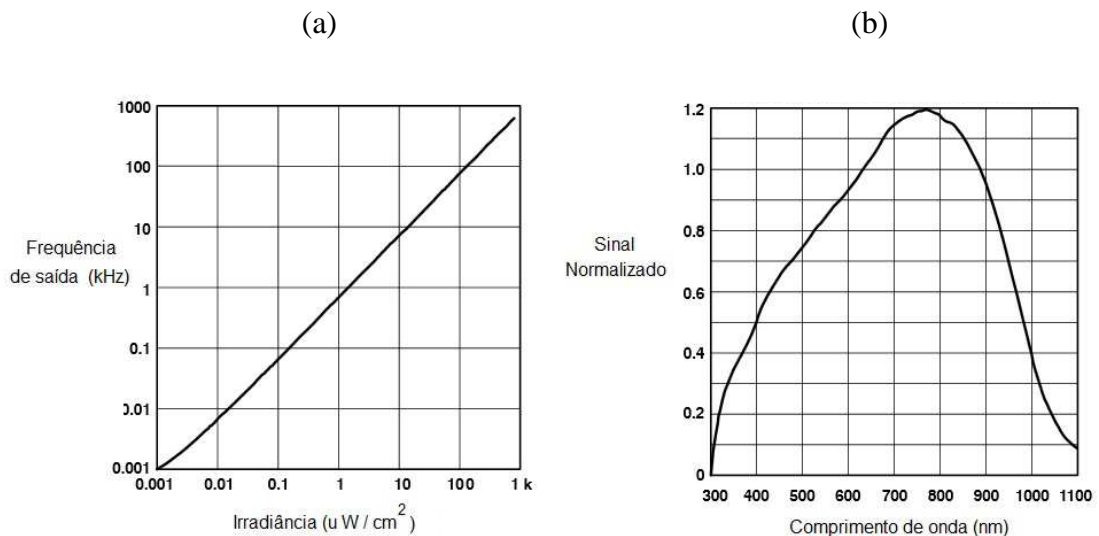
Para a realização dos experimentos foi projetado um dispositivo para uso *in-situ*, construído em plástico ABS através da técnica de impressão 3D, usando impressora do tipo REPRAP (Metamáquina, Brasil), o qual serviu de suporte para os emissores lasers com emissão de radiação luminosa nos comprimentos de onda de 530 nm e 660 nm, e receptores baseados no sensor TAOS TSL235R, também construídos especificamente para o transcorrer do trabalho. Os receptores e emissores foram montados dentro de tubos de Tecnyl® selados. O conjunto pode ser observado na Figura 2.

Figura 2 – (a) Projeto da Peça; (b) Suporte Pronto com Emissores e Receptores



Os emissores lasers usados foram baseados em semicondutor, de potência de 5 mW, respectivamente de 530 nm e 660 nm. O receptor óptico usado foi o sensor TSL235R (TAOS Inc.) - conversor de luz para frequência - o qual combina um fotodiodo de silício e um conversor de corrente para frequência em um circuito integrado CMOS monolítico. A saída é uma onda quadrada (50% de ciclo ocupado) com frequência diretamente proporcional à intensidade luminosa no fotodiodo. A caracterização da resposta do sensor quanto ao sinal de saída em 635 nm e a curva de correção de responsividade espectral mostrada na Figura 3.

Figura 3 – (a) Sinal dos Sensores; (b) Responsividade Espectral (TAOS INC. – 2007)



Conforme mostra a Figura 3 a resposta linear do sensor escolhido está na faixa de 1×10^{-3} kHz até 1×10^3 kHz, para um sinal de irradiância de entrada variando entre 1×10^{-3} $\mu\text{W}/\text{cm}^2$ até $1000 \mu\text{W}/\text{cm}^2$, com responsividade espectral da ordem de 80% para o comprimento de onda de 530 nm e de 100% para o comprimento de onda de 660 nm.

A saída digital direta como um sinal de frequência permite a interface direta com microcontrolador. A coleta de dados foi realizada pelo uso da interface serial do microcontrolador Arduino Due (32 bits), há uma taxa de 115200 bits/s. O programa de acionamento dos lasers e das leituras dos sensores foi feito em C++.

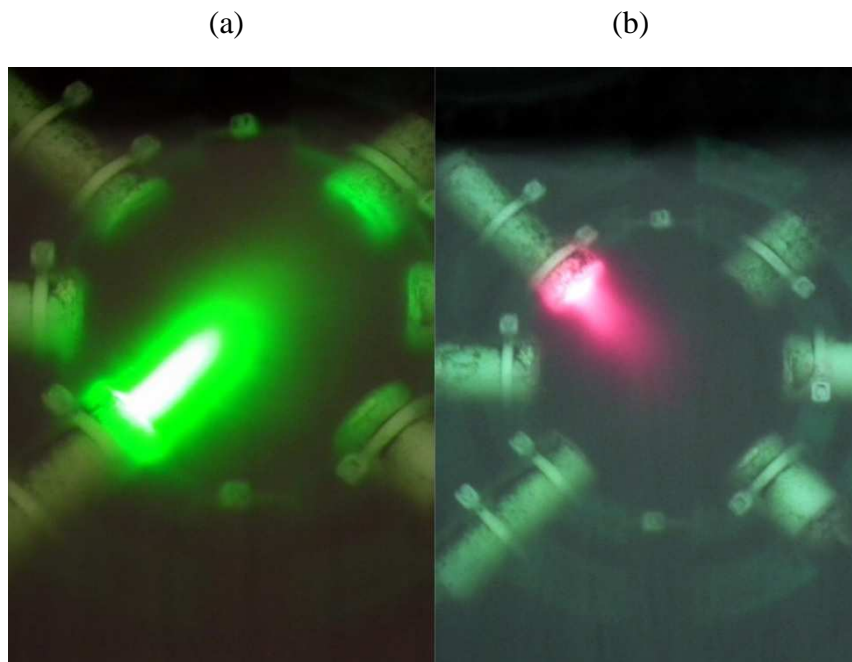
O processo de teste e calibração do sistema óptico desenvolvido foi realizado por diluições sucessivas no interior da câmara de suspensão microalgal do sistema de calibração. Este processo iniciou com a utilização de uma suspensão de *Spirulina sp* Leb 18, a qual foi oriunda de cultivo em *raceway outdoor*, devidamente concentrada em centrífuga, até 2,48 g/L, sendo que o procedimento experimental de medição do espalhamento da radiação luminosa foi aplicado em diferentes ângulos entre emissor e receptor. Após a coleta dos dados do primeiro ponto experimental, uma porção da suspensão foi retirada e o volume do sistema foi completado com o mesmo volume de meio de cultivo e o processo de obtenção das medidas de espalhamento foi repetido.

Cada ponto de concentração foi aferido através da massa seca de biomassa no ponto inicial e em cada diluição. Esta análise foi realizada através de uma adaptação da metodologia proposta por Zittelli e colaboradores (2000). Uma porção da suspensão de células era então filtrada através da membrana em frasco kitasato sob o efeito de uma bomba de vácuo para facilitar o processo. O volume de solução a ser filtrado foi ditado por uma aproximação empírica da concentração da solução. A partir da concentração estimada o volume foi calculado de modo que a diferença de massa final entre a membrana antes e após a filtração fosse de 0,01 g. A membrana foi então seca por 3h a 100°C e o quociente entre a diferença de massa antes e após a filtração e o volume de solução filtrado gera o valor de biomassa da diluição. A análise foi realizada no mínimo em duplicata para cada ponto de calibração.

3. RESULTADOS E DISCUSSÃO

A Figura 4 mostra o espalhamento da radiação luminosa na suspensão de *Spirulina* sp. LEB 18, quando sujeita ao comprimento de onda de 530 nm e ao de 660 nm, conforme ocorriam os experimentos.

Figura 4 – Aspecto do Espalhamento da Luz na Suspensão, (a) em 530 nm e (b) em 660 nm.



A interpretação do fenômeno físico associado com a passagem da radiação monocromática através da suspensão de microalgas é complexa, pois envolve efeitos de absorção, dispersão, transmissão, e emissão de radiação por células individuais, e conjuntos de células, gerando espalhamento por múltiplos percursos (HULST, 1981).

A presença de substâncias químicas no meio extracelular também interfere, especialmente pela absorção da radiação pelos pigmentos, carotenoides, alfa e beta-clorofila (GRIFFITHS, 2011). Por isso o uso de um comprimento de onda de 530 nm, onde a absorção da radiação foi significativamente menor, para vários teores de clorofilas, segundo Gitelson e colaboradores (1996).

O uso da fonte de 660 nm de comprimento de onda se justificava, porque esse é o comprimento de onda usado em medidas de densidade ótica, segundo a metodologia usada por Costa e colaboradores (2002).

Durante a realização dos experimentos, a utilização do comprimento de onda de 660nm mostrou-se inapropriado, devido à forte absorção da radiação nesse comprimento de onda, para as faixas de concentração avaliadas. Esse comportamento se justifica pois na metodologia de Costa et al. (2002), para medição da concentração de biomassa pela medição da densidade óptica, há o procedimento de diluição da amostra para a realização das medidas e confecção da curva-padrão, que permite o uso da lei de Lambert-Beer, enquanto que na medição *in situ* não há essa diluição. A partir dessa interferência dos pigmentos, para as faixas de concentração utilizadas, os experimentos usando a emissão em 660 nm não prosseguiram.

A Tabela 1 apresenta os resultados das medidas de intensidade de sinal nos receptores ópticos, para o comprimento de onda de 530 nm.

Da análise dos resultados mostrados na Tabela 1 observa-se que para os ângulos de 45° e 90° o sinal apresenta um comportamento crescente conforme aumenta a concentração e de acordo com a literatura (HUSTL,1981; BOHREN & HUFFMAN, 1983). Esse resultado é previsto, pois o espalhamento da radiação nessas direções aumenta conforme aumenta o número de partículas por unidade de volume, e também devido a razão entre o tamanho médio das células e o comprimento de onda, conforme (ZANGWILL, 2012).

Na faixa de concentração até 0,726 g/L o feixe luminoso satura o sensor, no ângulo de 180°. Devido a essa necessidade de satisfazer a condição de não saturar o sensor óptico, o uso do ângulo de 180° entre emissor e receptor fica limitado para concentrações acima de 1 g/L. Desta maneira um analisador de concentração de biomassa construído com fonte de 530 nm de 5mW de potência baseado somente em absorção, a priori, fica limitado a uma faixa de concentração alta, para as mesmas condições de projeto do equipamento.

Para o ângulo de 180° o comportamento é totalmente diverso, com a intensidade do sinal diminuindo conforme aumenta a concentração, devido a dominar o efeito de absorção, ou extinção da radiação sob o efeito do espalhamento, também justificado de acordo com Hustl (1981).

Para o ângulo de 135° o comportamento é fortemente não linear, tendo um máximo de sinal ao redor da concentração de 0,527 g/L.

Tabela 1 - Sinal no Sensor Luminoso para 530 nm em Função da Concentração e do Ângulo entre Emissor e Receptor (Médias e Desvios Padrões de 10000 Amostras)

Concentração (g/L)	Sinal (kHz)			
	45°	90°	135°	180°
0,000	8,040 ± 0,780	4,974±0,583	21,765±1,516	-
0,228	6,392±0,585	7,478±0,566	61,704±5,145	-
0,527	6,946±0,404	13,334±0,725	75,535±2,980	-
0,726	9,085±0,892	15,786±0,969	66,852±2,582	-
0,967	11,991±0,646	19,535±0,994	65,842±2,961	143,515±2,028
1,203	12,142±0,710	20,874±0,934	52,574±2,245	121,217±4,804
1,440	14,376±0,941	22,658±1,413	43,416±1,452	69,241±2,666
1,950	14,974±1,460	20,660±1,365	29,645±1,873	43,754±2,775
2,480	21,892±1,651	22,216±1,121	21,707±1,116	29,426±2,044

Com o intuito de avaliar a dispersão das medidas, foi usado o desvio padrão relativo (SKOOG et al. , 2001; WERNIMONT, 1985), o qual foi significativo, em algumas medições chegando próximo de 10%. Essa dispersão das medidas foi associada com o projeto do sistema de ativação do laser e posterior medição da radiação, pois de início não havia sido considerada a necessidade de tempo para o equipamento construído estabilizar a leitura, após a ativação do circuito elétrico de alimentação do emissor laser.

Foram realizadas novas medidas (Tabela 2), com tempo de espera entre o acionamento do laser e a realização das leituras nos sensores ópticos de 10s, para garantir a estabilização da voltagem aplicada sob o emissor. Como esperado, o desvio padrão percentual

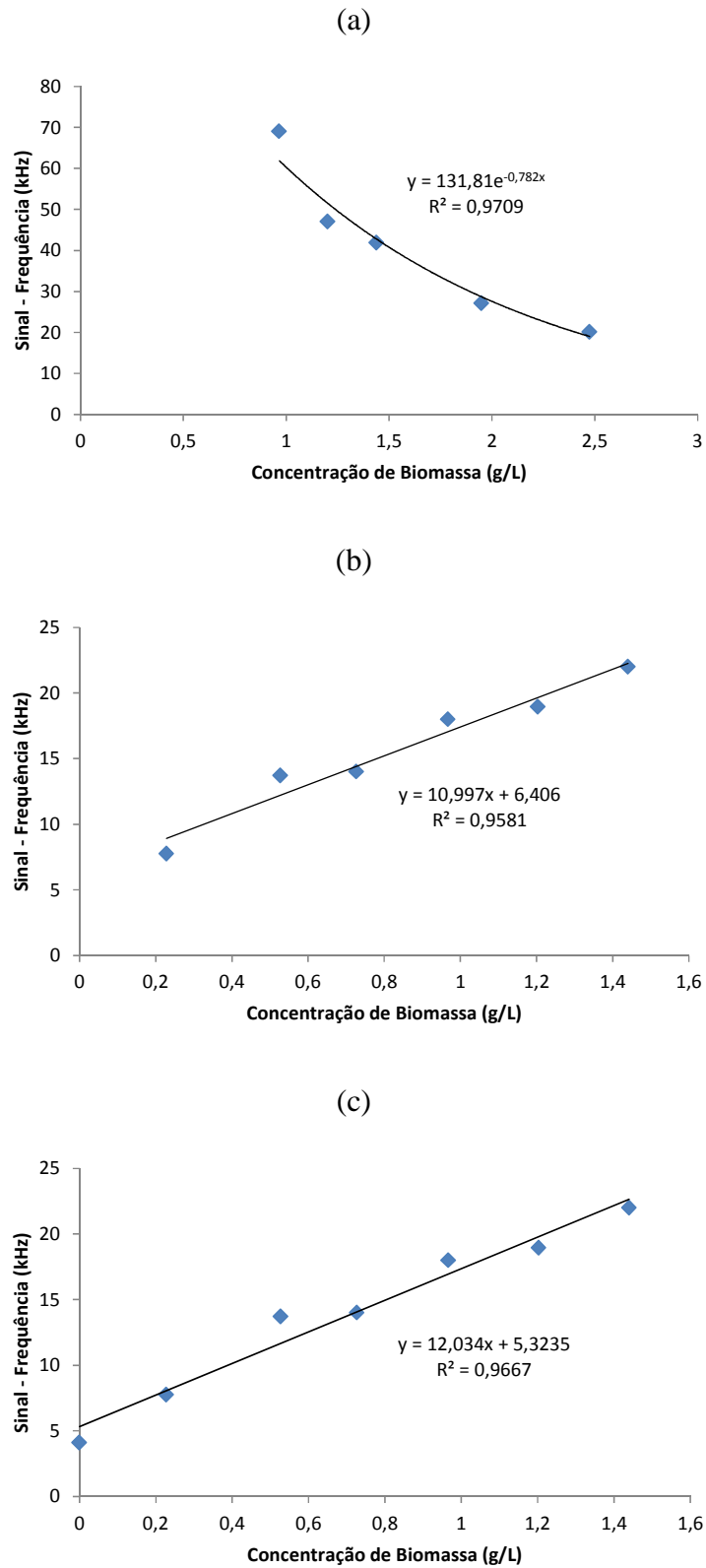
global caiu, apresentando valor máximo de 7,6%, mas ficando em média próximo de 1,9%.

Tabela 2 - Sinal no Sensor Luminoso para 530 nm em Função da Concentração e do Ângulo entre Emissor e Receptor (Médias e Desvios Padrões de 2500 Amostragens)

Concentração (g/L)	Sinal (kHz)			
	45°	90°	135°	180°
0,000	7,075 ± 0,263	4,110±0,312	20,037±0,189	-
0,228	5,919±0,272	7,757±0,546	50,742±0,629	-
0,527	6,045±0,207	13,708±0,659	69,062±0,369	-
0,726	7,127±0,333	14,017±0,133	59,220±0,541	-
0,967	11,934±0,248	18,012±0,110	68,165±0,481	144,456±1,046
1,203	10,994±0,082	18,941±0, 235	47,006±0,118	115,947±0,249
1,440	13,089±0,284	21,989±0,162	41,874±0,331	63,132±0,339
1,950	13,069±0,254	19,038±0,191	27,110±0,313	40,945±0,227
2,480	19,997±0,133	21,130±0,336	20,142±0,350	27,050±0,223

Foram avaliadas as correlações entre as intensidades de sinal, em kHz, e as concentrações, para os ângulos de 135° e 45°, conforme as Figuras 5 e 6, respectivamente.

Foram avaliadas as correlações entre as intensidades de sinal, em kHz, e as concentrações, para os ângulos de 135°, 45° e 90°, conforme as Figuras 7a, 7b, e 7c respectivamente.

Figura 5 – Sinal no Sensor TAOS e Concentrações de Biomassa- (a) 135°, (b) 45° e (c) 90°

Os coeficientes de determinação (R^2) entre sinal e concentração determinados foram sempre maiores do que 0,95. Há a possibilidade de abranger uma faixa bastante ampla de concentração de biomassa, de 0,2 g/L até 2 g/L, desde que sejam combinados dois sensores ópticos para o mesmo emissor, para varrer toda a faixa de concentração da suspensão da microalga *Spirulina*.

A sensibilidade média nos experimentos de espalhamento, definida como sendo a derivada do sinal no sensor com respeito à concentração, para os ângulos de 45° e 90° foi, respectivamente, de 11,0 kHz/(g.L⁻¹) e de 12,0 kHz/(g.L⁻¹), enquanto que para o ângulo de 135°, a sensibilidade, mínima, para fins de comparação foi de 13,2 kHz/(g.L⁻¹).

A correlação entre o espalhamento da radiação e a concentração mássica em cultivo de microalgas, Figura 5, é corroborada pelo trabalho de Gitelson e colaboradores (1995) onde foram avaliadas e correlacionadas a refletância e a atenuação vertical, na faixa de 400 até 1100 nm, com a densidade microalgal, com o intuito de criar algoritmos para a estimativa remota da concentração de biomassa.

As diferenças de comportamento observados entre os espalhamentos da radiação monocromática pela suspensão de *Spirulina*, em função dos ângulos entre emissor e receptor, bem como em função da concentração da biomassa estão de acordo com o esperado teoricamente, conforme Mishchenko e colaboradores (2002), devido à natureza morfológica das células, quanto ao seu formato, e quanto ao tamanho médio das mesmas, com respeito ao comprimento de onda da radiação monocromática utilizada.

4 CONCLUSÕES

O conhecimento da concentração de biomassa nos cultivos de microalgas é importante parâmetro para efetivar a operação da unidade produtiva, especialmente a fim de possibilitar a determinação da política de operação do biorreator, ao longo do ciclo de produção.

A determinação da existência de correlação estatística ($R^2 > 0,95$) entre o sinal lido pelo receptor TAOS TSL235R, durante o espalhamento da radiação de 530 nm, especificamente para os ângulos entre emissor e receptor de 45°, 90° e 135°, na faixa de concentração da suspensão de *Spirulina* entre 0,2 g/L até 2,0 g/L, reafirma a possibilidade de

desenvolvimento de sensores ópticos de determinação de biomassa, para uso *in situ*, sem necessidade de diluição.

As medições de espalhamento da radiação monocromática em 530 nm pela suspensão de *Spirulina* que tiveram maior sensibilidade, na faixa entre 0,2 g/L e 1,4 g/L foram aquelas com um ângulo entre emissor e receptor de 90°, sendo 9,1% em média maiores do que com o ângulo de 45°.

A partir do projeto e da elaboração de um conjunto de medição da intensidade da radiação luminosa monocromática absorvida e/ou espalhada pela suspensão de células, usando hardware desenvolvido especificamente para o fim da determinação de concentração da *Spirulina* sp. LEB 18, foi possível vincular as características específicas de espalhamento e as consequentes medidas realizadas com a concentração do micro-organismo na suspensão, o que possibilitará o uso futuro do sensor óptico desenvolvido na produção em larga escala da biomassa microalgal.

5 REFERÊNCIAS

- ANDRADE, M. da R.; CAMERINI, F. V.; COSTA, J. A. V. Chemical carbon losses and growth kinetics in *Spirulina* cultures. *Química Nova*, v. 31, n. 8, p. 2031–2034, jan. 2008.
- BANZI, M.; *Getting Started with Arduino*, O'Reilly Media, 2009.
- BEUERMANN, T.; EGLY, D.; GEOERG, D.; KLUG, K. I.; STORHAS, W.; METHNER, F.J.; On-line carbon balance of yeast fermentations using miniaturized optical sensors. PMID: 22100900: *Journal of Bioscience and Bioengineering*, v. 113, n. 3, p. 399–405, mar. 2012.
- BOHREN, C.F.; HUFFMAN, D.R.; *Absorption and Scattering of Light by Small Particles*; John Wiley, New York, 1983.
- CASTELIJN, A.A.; BENTHEM, R.C.; ASSEM, D.; Self compensating real-time biomass sensor. *Proceedings of the First International Symposium on Microgravity Research & Applications in Physical Sciences & Biotechnology*, Sorrento, Italy, 2000.
- COGNE, G., LASSEUR, C.H., CORNET, J.F., DUSSAP, C.G., GROS, J.B.; Growth monitoring of a photosynthetic micro-organism (*Spirulina platensis*) by pressure measurements. *Biotechnol. Lett.* 23, 1309–1314, 2001

COSTA, J.A.V.; COLLA, L.M.; DUARTE FILHO, P.; KABKE, K.; WEBER, A. Modeling of *Spirulina* platensis growth in fresh water using response surface methodology. World J. Microb. Biot., v. 18, p. 603-607, 2002.

COX, R. P.; MILLER, M.; NIELSEN, J.B.; NIELSEN, M.; THONSON, J. K.; Continuous turbidometric measurements of microbial cell density in bioreactors using a light emitting diode and a photodiode. Journal of Microbiological Methods 10, 25-31,1989.

GERTZ, E.; JUSTO, P.D.; Atmospheric Monitoring with Arduino: Building Simple Devices to Collect Data about the Environment, O'Reilly Media, 2012.

GITELSON, A. A. *et al.* Optical Properties of Dense Algal Cultures Outdoors and Their Application to Remote Estimation of Biomass and Pigment Concentration in *Spirulina* platensis (cyanobacteria), Journal of Phycology, v. 31, n. 5, p. 828–834, 1995.

GITELSON, A.; QIANG, H.; RICHMOND, A. Photic Volume in Photobioreactors Supporting Ultrahigh Population Densities of the Photoautotroph *Spirulina* platensis. PMID: 16535309 PMCID: PMC1388847: Applied and Environmental Microbiology, v. 62, n. 5, p. 1570–1573, 1996.

GRIFFITHS, M. J. *et al.* Interference by pigment in the estimation of microalgal biomass concentration by optical density. PMID: 21329736: Journal of microbiological methods, v. 85, n. 2, p. 119–123, 2011.

HULST, H. C.; Light Scattering by Small Particles, Dover Books on Physics, 1981.

MAROSE, S.; LINDEMANN, C.; ULBER, R.; SHEPER, T.; Optical sensor systems for bioprocess monitoring. Trends in Biotechnology, v. 17, n. 1, p. 30–34,1999.

MEIRELES, L. A. *et al.* On-line control of light intensity in a microalgal bioreactor using a novel automatic system. Enzyme and Microbial Technology, v. 42, n. 7, p. 554–559, 2008.

MIE, G. von; Beiträge zur Optik trüber Medien, speziell kolloidaller Metallösungen. Annalen der Physik, n° 4, vol 25, p. 377-445, 1908.

MISHCHENKO, M. I.; TRAVIS, L. D.; LACIS, A.A.; Scattering, Absorption, and Emission of Light by Small Particles. Cambridge University Press, 2002.

SKOOG, D. A.; HOLLER, F.J.; NIEMAN, T.A.; Principios de Análisis Instrumental; McGraw-Hill Madrid, 5a. Ed.; 2001.

TREVISAN, M. G.; POPPI, R. J. Process Analytical Chemistry. Química Nova, v. 29, n. 5, p. 1065–1071, 2006.

VOJINOVIĆ, V.; CABRAL, J. M. S.; FONSECA, L. P. Real-time bioprocess monitoring: Part I: In situ sensors. Sensors and Actuators B: Chemical, v. 114, n. 2, p. 1083–1091, 2006.

SHULER, M.L.; KARGI, F.; Bioprocess Engineering: Basic Concepts, 2a. Ed., Prentice Hall, 2001.

WERNIMONT, G. T.; use of Statistics to Develop and Evaluate Analytical Methods, Edited by W. Spendley, AOAC, Arlington, Virginia, USA, 1985

ZANGWILL, A.; Modern Electrodynamics, Cambridge University Press, 2012.

ZARROUK, C. Contribution à l'étude d'une Cyanophycée: influence de divers facteurs physiques et chimiques sur la croissance et la photosynthèse de *spirulina máxima*. 1966. Tese (Ph.D) - Université Des Paris, Paris, 1966.

ZITTELLI, G. C., PASTORELLI R., TREDICI M. R. A modular flat panel photobioreactor (MFPP) for indoor mass cultivation of *Nannochloopsis sp.* under artificial illumination. Journal of Applied Phycology, v. 12, p 521 – 526, 2000.

ARTIGO III: Sensor Óptico Combinado Definido por Software para Determinação da Concentração de *Spirulina* sp. LEB 18.

RESUMO

Métodos instrumentais de análise fazem uso de propriedades físico-químicas que são correlacionadas com determinada quantidade de um analito, através de uma curva de calibração. As classes de métodos ópticos de análise, conhecidos como espectrofotométricos e fotocolorimétricos, baseados na medição da densidade óptica, estão fundamentados na Lei de Lambert-Beer, que tem validade para soluções diluídas, e desconsidera o espalhamento da radiação luminosa incidente no meio, provocado pela interação entre partículas suspensas de tamanho muito maior do que o comprimento de onda usado na medição. Este trabalho teve por objetivo modelar um sensor óptico combinado definido por software para determinação da concentração de biomassa de *Spirulina* sp. LEB 18, em suspensão no meio de cultivo. O sensor desenvolvido é dito combinado, porque faz uso de duas informações de intensidade de radiação, em dois ângulos de incidência diferentes. Com o intuito de identificar o padrão de atenuação da radiação luminosa com as concentrações variáveis de biomassa foi usada uma rede *neurofuzzy* do tipo Takagi-Sugeno-Kang (TSK), onde os parâmetros das funções de pertinência e coeficientes lineares das funções se-então do mecanismo de inferência foram determinadas por treinamento supervisionado. O método matemático usado no treinamento foi a minimização do somatório do erro absoluto, através do uso do algoritmo de otimização por enxame de partículas. Foram avaliadas as combinações 2 a 2 de medidas de sinal no sensor óptico obtidas para ângulos testados. A melhor fonte de informações para um soft sensor a partir do espalhamento da radiação monocromática em 530 nm foi o par 45°/135°. Foi obtido um erro médio menor do que 3% em concentração durante a validação do modelo, com redes que durante o treinamento, tinham com um coeficiente de determinação (R^2) de 0,99, para a concentração da suspensão de *Spirulina* entre 0,2 g/L até 2 g/L.

Palavras chave: química analítica de processos, rede *neurofuzzy*, sensor virtual

Software Aided Optical Sensor for *Spirulina* sp. LEB 18 Biomass Monitoring

ABSTRACT

A wide range of instrumental methods employ the correlation of a physical-chemical characteristic of a sample with a given compound standard curve in order to determine the compound presence in the sample. In the same manner, optical analysis, among them optical density, spectrophotometric and photometric methodologies, are founded in the Lambert-Beer law. However, this law is only valid in diluted solutions, and disregards light scattering inside the solution propitiated by particles with lengths similar to the radiation source wavelength. This work objective is to model a software layer in order to convert optical data from suspensions of *Spirulina* sp. LEB 18 into biomass concentration. The data originates from increasing know concentrations of a biomass suspension and measured radiation, originated from a laser emitter, by TAOS TSL235R receptor in angles of 45°, 90° and 135° from the emitter original axis. The sensor is combines the information from two different angles of receptors in order to generate the biomass concentration value. In order to recognize the scattering patterns from different biomass concentrations and receptor angles a Takagi-Sugeno-Kang neurofuzzy network was utilized. The network internal pertinence parameters, as well as the coefficients from the if/then were determinated through supervised learning. The numerical method in order to minimize the sum of absolute error between predicted and experimental values was a Particle Swarm Optimization algorithm. Combinations of two angles were tested in the training of the network. The best fitting occurred when the pair was the 45°/135°, for the 530nm wavelength emitter. The average error for this pair was less than 3% when compared to the experimental value and the network total coefficient of determination was 0,99.

Keywords: Neurofuzzy Network, Process Analytical Chemistry, Softsensor

1 INTRODUÇÃO

O uso de sensores de concentração de biomassa é fundamental no acompanhamento automatizado do cultivo de micro-organismos. A biomassa microalgal pode ter sua concentração ao longo do tempo determinada através da medida da densidade óptica em espectrofotômetro, mas em condições de processamento em larga escala são necessários métodos expeditos de acompanhamento, robustos e que possam ser integrados com a automação de processos.

A disponibilidade de sensores definidos por software permite um melhor monitoramento e consequente controle de bioprocessos, pois esses sensores incorporam modelos matemáticos decorrentes dos sistemas em estudo, sejam esses mecanísticos ou fenomenológicos. Os sensores definidos por software são úteis para a monitoração de bioprocessos, devido à complexidade dos mecanismos biológicos das células.

O desenvolvimento de sensor de concentração de biomassa combinando informações de dois ângulos de incidência entre emissor e receptor de radiação luminosa é uma alternativa, pois pode aumentar a faixa de concentração de utilização do sensor óptico, em condições de cultivo, além da região de validade da lei de Lambert-Beer. Para efetivar esse tipo de medição é necessário integrar as duas medidas, em dois ângulos de incidência diferentes, com as respectivas medidas de concentração.

O objetivo do presente trabalho foi desenvolver um sensor definido por software para o conjunto de medidas ópticas combinadas de espalhamento e absorção da radiação monocromática em 530 nm para uma suspensão da cianobactéria *Spirulina* sp. LEB 18.

O uso da lógica *fuzzy* e de “soft computing” em problemas de engenharia biológica e na agricultura já ocorre há mais de trinta anos (HUANG et al. ,2010), pois um conjunto de técnicas computacionais "inexatas" são propícias para modelar e analisar problemas complexos, oferecendo soluções com tolerância à imprecisão e incerteza, visto que a lógica *fuzzy* utiliza variáveis linguísticas multi-valor, não-numéricas para modelar o raciocínio humano aproximado (ZADEH, 1975). A tolerância à imprecisão e incerteza está vinculada a possibilidade de trabalhar na ausência de limites estritos (BAUGHMAN & LIU, 1996).

De acordo com Rashid & Ahmed (2012) o sistema de inferência adaptativo *neurofuzzy* (ANFIS) é um tipo de rede neural baseado no sistema de inferência *fuzzy* do tipo TSK - Takagi-Sugeno-Kang (SUGENO, 1985), o qual integra tanto princípios de redes neurais como de lógica *fuzzy*, com potencial de agregar os benefícios de ambos simultaneamente – a capacidade de realizar o “raciocínio aproximado” e de simultaneamente armazenar conhecimento através do modelo conexionista.

Os sistemas conexionistas baseados em redes neurais artificiais tem sido usados na automação de bioprocessos (GADKAR, 2005). Wali e colaboradores (2012), por exemplo, usaram a arquitetura de redes ANFIS com treinamento fundamentado em algoritmos genéticos para o controle online da temperatura em reator de produção de biodiesel usando micro-ondas.

Ao modelar a qualidade do efluente da digestão anaeróbia Erdirencelebi e Yalpir (2011) reafirmaram a capacidade de sistemas ANFIS em construir qualquer função não-linear, com múltiplas entradas e saídas, avaliando o desempenho do modelo desenvolvido por meio do coeficiente de correlação (R^2) e do erro percentual médio.

Modelos matemáticos de processos podem ser utilizados em conjunto com sensores para estimar variáveis de processo, que não podem ser medidas diretamente. Esta abordagem – baseada na teoria de controle automático - é amplamente utilizada, incluindo engenharia de bioprocessos; tais estimadores baseados em computação foram denominados “sensores de software” (*soft sensors*), refletindo o fato de que eles podem substituir ou apoiar sensores clássicos, baseados em hardware (HAVLIK et al., 2013). Outros autores, como Pitasse e Fileti (2001), adotam a nomenclatura de sensores virtuais.

Esses sensores fazem uso, de alguma forma, da estimação de estado, cujo objetivo é estimar as variáveis de estado de um sistema a partir de determinadas medições online/inline (LU, 1996). Em geral, as aplicações destas técnicas envolvem predição, filtragem e suavização de informações (GELB, 1974).

Embora os sistemas de sensores existentes para o cultivo de microalgas possam fornecer medições on-line para os parâmetros físicos de interesse em cultivos, sistemas de monitoramento de importantes variáveis químicas e biológicas não são tão bem desenvolvidos, o que ocorre devido ao elevado grau de complexidade das células e do seu ambiente químico (HAVLIK et al., 2013).

Redes neurais podem ser usadas como funções flexíveis de regressão não linear, pois propiciam um grande número de parâmetros de ajuste que podem ser usados para aproximar qualquer função (VELTEN, 2009).

Conforme Haykin (1994) o aprendizado é um processo no qual os parâmetros livres de uma rede neural são adaptados através de um processo contínuo de estimulação pelo ambiente no qual a rede está incorporada. O tipo de aprendizado é determinado pela maneira pela qual a mudança dos parâmetros ocorre.

Dentre os tipos de treinamento, o treinamento pela correção do erro faz uso de métodos de otimização com o intuito de minimizar uma função objetivo, definida como o sinal de erro, a diferença entre a resposta desejada, $d_k(n)$, e a resposta atual, $y_k(n)$.

A otimização pelo enxame de partículas (Kennedy e Eberhart, 1995; Kennedy Eberhart, 2001) é um paradigma computacional fundamentado no fenômeno da inteligência coletiva exibida por enxames de insetos, cardumes de peixes e bando de pássaros, que tem sido aplicada na solução de diversos problemas de otimização dos mais variados campos de pesquisa (Eberhart e Shi, 2001).

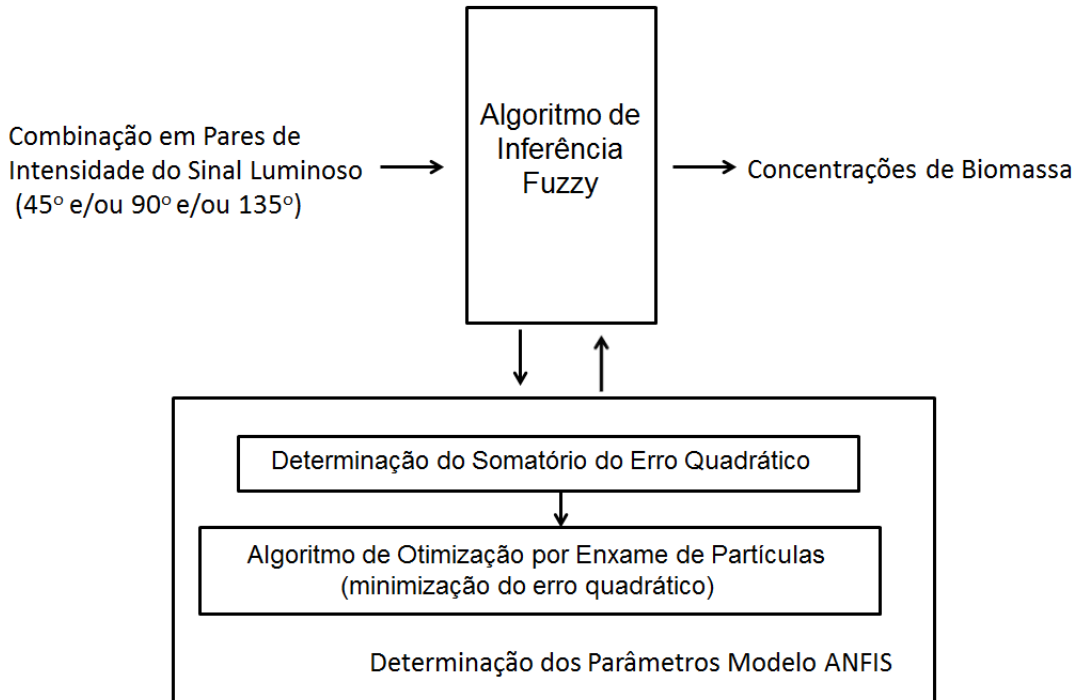
2. MATERIAIS E MÉTODOS

Para obter os dados de treinamento para a rede *neurofuzzy* (ANFIS) do sensor por software, compostos das informações sobre o espalhamento da radiação luminosa de 530 nm em várias concentrações de suspensões de biomassa microalgal de *Spirulina* sp. LEB 18, fez-se anteriormente o cultivo em escala semi-piloto em reatores abertos do tipo *raceway* cobertos por uma estufa de filme translúcido com proteção UV, usando o meio Zarrouk (ZARROUK, 1966), diluído em 50%, quanto à concentração da fonte de carbono (ANDRADE, 2008), em regime semi-contínuo.

Em suspensões de concentração variável de biomassa, determinada por análise de massa seca (ZITTELLI et al., 2000), fez-se medições da radiação luminosa espalhada pela suspensão de células, gerada num laser de 5 mW e de comprimento de onda de 530 nm usando o sensor TAOS TSL235R (TAOS INC) nos ângulos de 135° e 90° entre emissor e receptor.

Foi implementado no software Scilab 5.5 o programa de treinamento do sistema ANFIS conforme o diagrama mostrado na Figura 1.

Figura 1 – Arquitetura do Sistema ANFIS Utilizado.

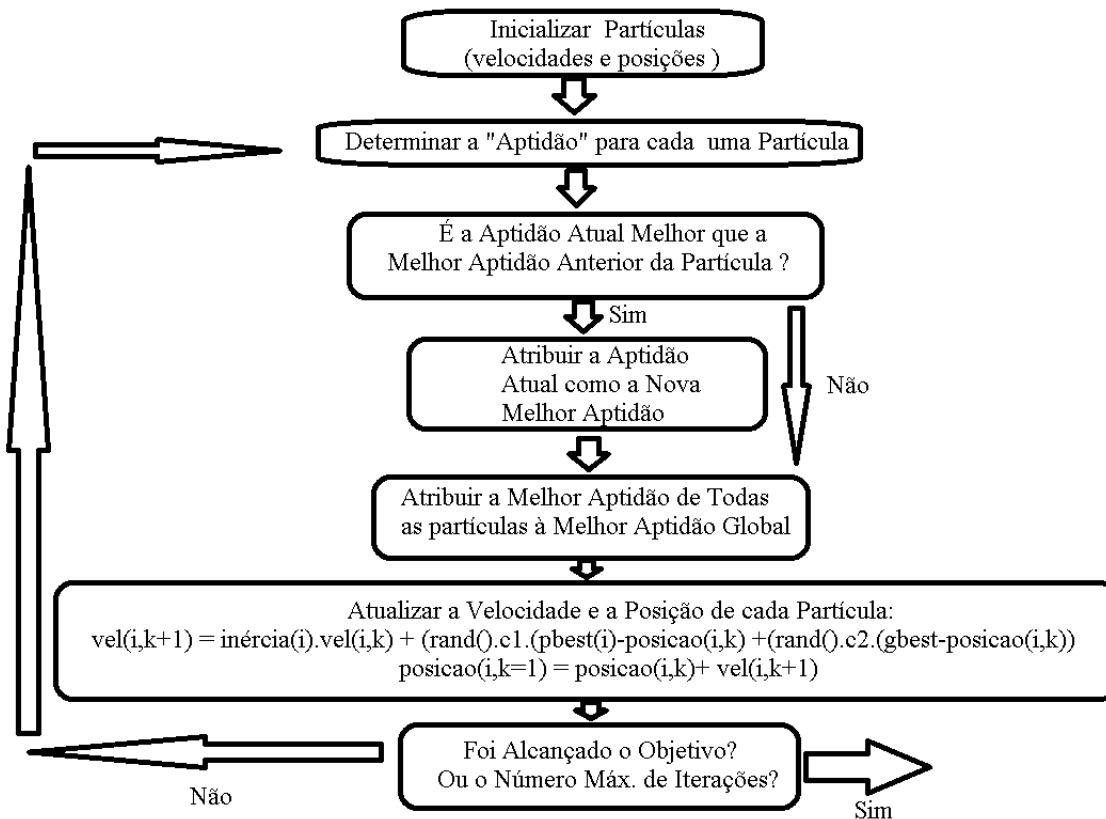


O conjunto de dados de treinamento foi normalizado, e particionado no espaço *fuzzy* de entrada, de forma a gerar a parte consequente das regras se-então da inferência *fuzzy*. Para cada regra houve uma função linear do tipo saída = pertinência(x). $A(i) + b(i) +$ pertinência(y). $A(i) + b(i)$. Onde a saída foi a concentração e x e y as intensidades de intensidade luminosa nos ângulos de espalhamento testados.

A ativação de cada regra foi feita de acordo com a norma min-max conforme (DRIANKOV, 1996). A saída do sistema TSK não precisa ser defuzzificada, visto que a mesma já trabalha no universo de discurso das variáveis de treinamento.

O algoritmo de minimização do somatório do erro absoluto, ou aptidão, utilizado é descrito na figura 2. O algoritmo de otimização por exame de partículas implementado foi adaptado para permitir a determinação dos parâmetros A_{iii} , B_{iii} , C_{iii} , D_{iii} correspondentes aos parâmetros linear e angular das equações lineares das saídas das regras se-então do sistema de inferência *fuzzy*, para cada regra *fuzzy*.

Figura 2 – Algoritmo da Otimização por Enxame de Partículas Utilizado



3. RESULTADOS E DISCUSSÃO

Os dados usados para treinamento do sensor por software são mostrados na Tabela 1. Para fins de realizar os treinamentos das redes ANFIS, houve a combinação 2 a 2 do conjunto de dados de treinamento, em função dos ângulos, resultando em 3 conjuntos de dados: pares 45°/90°, 90°/135 e 45° /135° de entrada, tendo a concentração de biomassa como saída do modelo.

Antes do treinamento as respectivas intensidades mensuradas em cada ângulo e as concentrações de biomassas são normalizadas.

Tabela 1 - Sinais dos Sensores Ópticos Função da Concentração e do Ângulo entre Emissor e Receptor, Usados no Treinamento do Sensor por Software (ANFIS).

Concentração (g/L)	Ângulo		
	45°	90°	135°
0,228	5,919±0,272	7,757±0,546	50,742±0,629
0,527	6,045±0,207	13,708±0,659	69,062±0,369
0,726	7,127±0,333	14,017±0,133	59,220±0,541
0,967	11,934±0,248	18,012±0,110	68,165±0,481
1,203	10,994±0,082	18,941±0,235	47,006±0,118
1,440	13,089±0,284	21,989±0,162	41,874±0,331
1,950	13,069±0,254	19,038±0,191	27,110±0,313
2,480	19,997±0,133	21,130±0,336	20,142±0,350

As configurações usadas para os parâmetros do algoritmo PSO usado no treinamento da rede são mostrados na Tabela 2.

Tabela 2 - Parâmetros do Algoritmo de Otimização por Enxame de Partículas (PSO)

Parâmetro	Valor
Número de Partículas	10
c1, c2	2,2
Inércias (máxima;mínima)	1; 0,2

Com a partição das entradas dos pares de ângulos com o intuito de realizar a “fuzzificação” necessitando da definição das funções de pertinência, as mesmas deveriam ser

também determinadas através do algoritmo de otimização, como parâmetros cujo o ajuste influencia na qualidade do ajuste do modelo. Essa opção, de funções de pertinências lineares determinadas pelo algoritmo, obrigaria a condição de trabalhar com uma otimização multiparamétrica com restrições, aumentando bastante o tempo de processamento, devido à necessidade de obedecer aos critérios de que o conjunto normalizado de sinais de entrada deveria obedecer aos limites entre [0;1]. Assim sendo, fez-se uso de funções de pertinência igualmente espaçadas: 0,25, 0,5 e 0,75 do conjunto normalizado de medidas de intensidade luminosa para o par de ângulos combinados.

Os resultados de treinamento estão apresentados na Tabela 3. É possível observar que em termos do erro absoluto médio o desempenho é razoável, e as faixas de exatidão alcançadas estavam bem próximas daquelas requeridas para a aplicação desejada: soft sensing de concentração de biomassa microalgal.

Tabela 3 - Avaliação dos Parâmetros de Representatividade do Modelo ANFIS+PSO com Funções de Pertinência Lineares Igualmente Espaçadas após 500 Iterações

Par de Ângulos entre Emissor e Receptor	Somatório do Erro Absoluto	Erro Médio	R ²
45° /90°	0,214	2,7%	0,980
45° /135°	0,236	2,95%	0,993
135°/90°	0,544	6,79%	0,960

O algoritmo PSO mostrou-se muito sensível ao ajuste fino das condições iniciais, em tempo de desenvolvimento, bem como ao tipo de função de ativação da regra *fuzzy* usado.

Apesar de recomendado como alternativa para modificação no algoritmo PSO original, foi observado que a limitação da velocidade das partículas provoca a limitação do comportamento inteligente emergente do enxame de partículas, pois reduz a possibilidade de deslocamento no espaço de busca, mesmo sabendo que velocidades muito altas provocam deslocamentos “aos saltos” que podem prejudicar a busca pelo ótimo global.

A solução de usar a inércia variável linearmente ao longo das iterações, entre uma inércia máxima e uma mínima, bem como usar uma inércia diferente para cada partícula,

parte da analogia com as diferentes massas que indivíduos de enxames reais apresentam. Essa diferenciação aumenta a diversidade do enxame, e mostrou-se útil no funcionamento do algoritmo.

A partir da análise do comportamento do erro absoluto médio ao longo das iterações para os 3 treinamentos, conforme mostrado na Figura 3, observa-se que não há o indício de que estivesse ocorrendo a chamada convergência prematura, descrita na literatura, quando da aplicação do algoritmo clássico PSO de Eberhart e Kennedy, para problemas multivariáveis, fruto de que o enxame cairia em uma “armadilha” de um mínimo local. Essa condição foi evitada, a partir da identificação de um número razoável de iterações, o enxame é realocado, preservando as informações das melhores posições de cada partícula ($pbest(i)$) e da melhor posição global ($gbest$).

O decaimento ao longo das iterações do somatório do erro absoluto para os 3 treinamentos, mostrado na Figura 3, foi bastante semelhante, mas cabe avaliar o desempenho do modelo quando comparado com os dados experimentais. A Figura 3 representa o conjunto de saídas de cada treinamento, conforme mostrado na tela para o usuário que fizer a utilização do software de treinamento da rede ANFIS + TSK + PSO, ao longo das iterações.

Da mesma forma a Figura 4, que mostra a saída do modelo em comparação com as medidas experimentais de biomassa de *Spirulina*, também está de acordo com a saída de tela do software desenvolvido em Scilab 5.5.

Figura 3 – Somatório do Erro Absoluto ao Longo das Iterações Durante o Treinamento da Rede ANFIS – TSK – PSO para os Pares Combinados de Ângulos

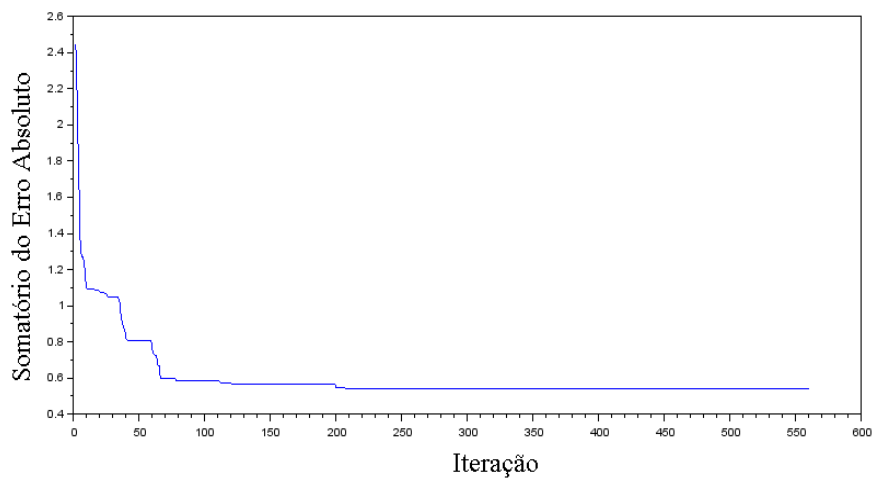
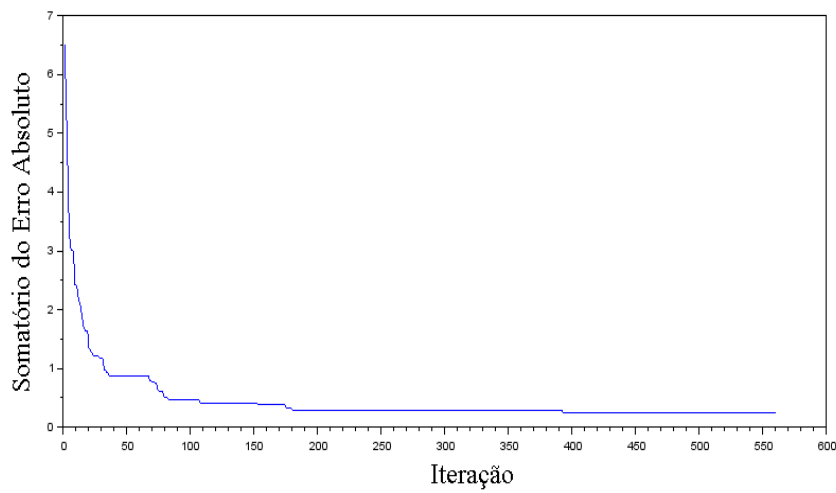
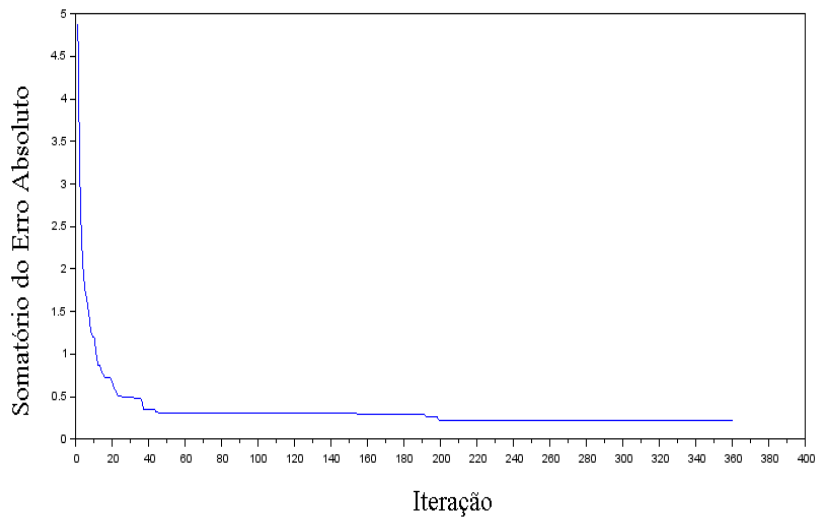
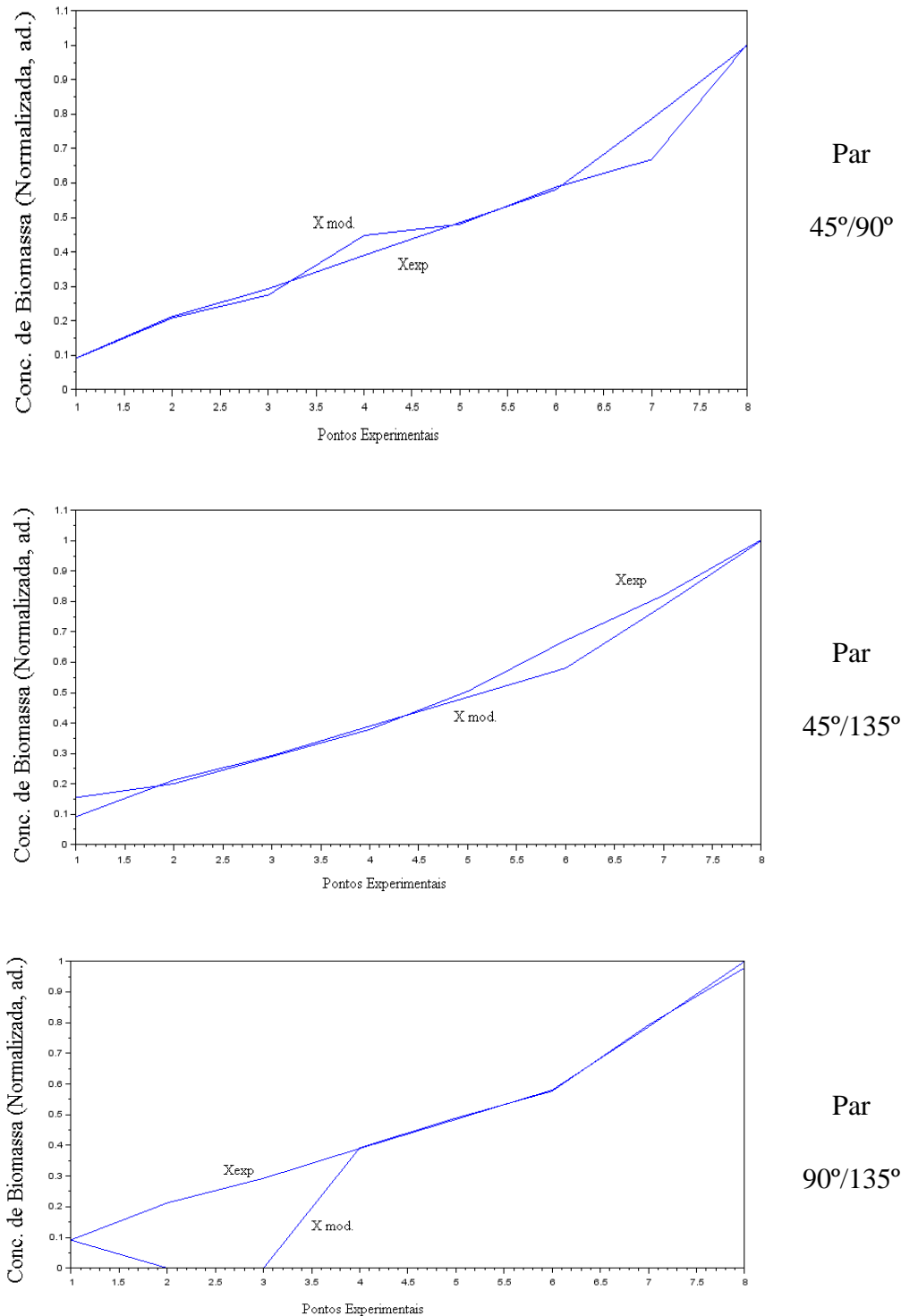


Figura 4 – Ajuste Após as Iterações e Treinamento da Rede ANFIS – TSK – PSO para os Pares Combinados de Ângulos – Conc. Experimental (X_{exp}) e do Modelo (X_{mod}).

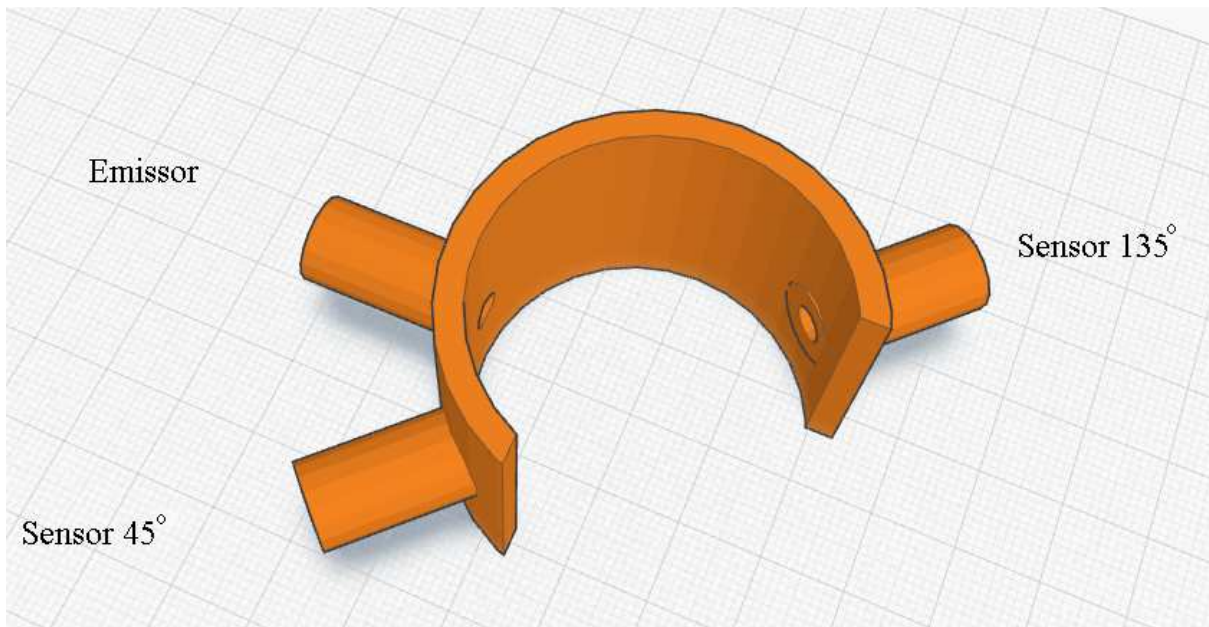


Da observação da Figura 4 se observa que exceto para o par 90°/135° houve representatividade dos modelos testados de sensor por software, com rede ANFIS-TSK, para todos os pontos experimentais de concentração. A partir da análise conjunta da Tabela 3, o

melhor modelo foi efetivamente aquele do par 45°,135°. A faixa de concentração de biomassa de *Spirulina* variando entre 0,228 g/L e 2,48 g/L, representa uma dificuldade maior para que o modelo possa representar o comportamento do espalhamento da radiação monocromática em 530 nm na suspensão da microalga.

A partir dos resultados dos experimentos foi concebido o projeto de um sensor combinado passível de ser impresso em uma impressora 3D, conforme a Figura 5.

Figura 5 – Concepção do Sensor de Biomassa Óptico Combinado



4. CONCLUSÕES

Foi desenvolvido um sensor definido por software para o conjunto de medidas ópticas combinadas de espalhamento da radiação monocromática em 530 nm para uma suspensão da cianobactéria *Spirulina* sp. LEB 18.

O melhor resultado de modelo para sensor por software foi obtido com um erro médio menor do que 3% durante o treinamento, com um coeficiente de correlação (R^2) 0,99 para a rede treinada com os pares de medidas de espalhamento em 45°,135°.

Foi concebido o projeto conceitual de um sensor óptico de biomassa combinado, com ângulos de 45° e 135° entre emissor e sensores, passível de ser impresso usando impressão 3D em plástico ABS ou PLA.

A possibilidade de utilização do sensor de software de concentração de biomassa em plantas pilotos ou de maior escala, usando a arquitetura desenvolvida é promissora, não só pela qualidade das informações, mas principalmente pela possibilidade de desenvolver uma tecnologia de sensores inline de concentração de baixo custo, robusta, passível de ser usada em plantas de larga escala de produção de biomassa de *Spirulina* sp. LEB 18.

5 REFERÊNCIAS

ANDRADE, M. da R.; CAMERINI, F. V.; COSTA, J. A. V. Chemical carbon losses and growth kinetics in *Spirulina* cultures. *Química Nova*, v. 31, n. 8, p. 2031–2034, jan. 2008.

BAUGHMAN, D.R; LIU, Y.A.; *Neural Networks in Bioprocessing and Chemical Engineering*, Academic Press, 1996.

DRIANKOV, D., HELLENDORN, H., REINFRANK, M., *An Introduction to Fuzzy Control*. 2. Ed. , Springer-Verlag, New York, 316 p, 1996.

EBERHART, R.C.; SHI, Y.; *Particle Swarm Optimization: Developments, applications and Resources*, Proc. Congress on Evolutionary Computation, IEEE, 2001.

ERDIRENCELEBI, D., YALPIR, S.; Adaptive network fuzzy inference system modeling for the input selection and prediction of anaerobic digestion effluent quality, *Applied Mathematical Modelling*, v.35, 8, pp 3821–3832, 2011.

GADKAR, K.G.; MEHRAB, S.; GOMES, J.; On-line adaptation of neural networks for bioprocess control, *Computers & Chemical Engineering*, Volume 29, Issue 5, 15, p.1047–1057, 2005.

GELB, A.; *Applied Optimal Estimation*, The MIT Press, 1974.

GODJEVAC, J. Comparison Between PID and Fuzzy Control. Internal Report R95.361. Ecole Polytechnique Fédérale, Lausanne, 1995.

HAYKIN, S.; *Neural Networks – A Comprehensive Foundation*, Macmillan Publishing Company, USA, 1994.

HAVLIK, I.; LINDNER, P.; SCHEPER, T.; REARDON, K.F.; On-line monitoring of large cultivations of microalgae and cyanobacteria, *Trends in Biotechnology*, Vol. 31, No. 7, July 2013.

HUANG, Y.; LAN, Y.; THOMSON, S.J.; FANG, A.; HOFFMAN, W.C.; LACEY, R. E.; Development of soft computing and applications in agricultural and biological engineering, *Computers and Electronics in Agriculture* 71, 107–127, 2010.

KENNEDY J., EBERHART, R.C. Particle Swarm Optimization. Proc. IEEE International Conference on Neural Networks, IEEE Service Center, Piscataway, NJ, IV: 1942-1948, Perth, Australia, 1995.

KENNEDY, J.; EBERHART, R.C.; *Swarm Intelligence*. Morgan Kaufmann. ISBN 1-55860-595-9, 2001.

LU, Y.; *Industrial Intelligent Control – Fundamentals and Applications*, John Wiley and Sons, 1996.

PITASSE, A.; FILETI, A. M. F.; *Redes Neurais em processos Siderúrgicos; Analisador Virtual de Propriedades metalúrgicas do Sínter e Modelo de Predição de Qualidade do Aço*, Tese de Doutorado, FEQ, UNICAMP, 2001.

RASHID, T. A.; AHMED, H.A.; Building an Adaptive Neural Fuzzy Inference System for Revised General Test/Gross Point Average Problem, *International Journal of Emerging Technologies in Computational and Applied Sciences-IJETCAS*, 12-326, 2012.

SUGENO, M., *Industrial applications of fuzzy control*, Elsevier Science Pub. Co., 1985.

VELTEN, K.; *Mathematical Modeling and Simulation: Introduction for Scientists and Engineers*, Wiley-VCH, 2009.

WALI, W.A.; AL-SHAMMA´A, A.I.; HASSAN, K.H.; CULLEN, J.D.; Online genetic-ANFIS temperature control for advanced microwave biodiesel reactor, *Journal of Process Control* 22, 1256– 1272, 2012.

ZADEH, L.A.; The concept of a Linguistic Variable and its Application to Approximate Reasoning – I, *Information Sciences* 8,199-299, 1975.

ZARROUK, C. Contribution à l'étude d'une Cyanophycée: influence de divers facteurs physiques et chimiques sur la croissance et la photosynthèse de spirulina máxima. 1966. Tese (Ph.D) - Université Des Paris, Paris, 1966.

ZITTELLI, G. C., PASTORELLI R., TREDICI M. R. A Modular Flat Panel Photobioreactor (MFPP) for Indoor Mass Cultivation of *Nannochloopsis sp.* Under Artificial Illumination. *Journal of Applied Phycology*, v. 12, p 521 – 526, 2000.

ARTIGO IV: Microcontroladores Arduino na Modelagem Cinética de Bioprocessos Autotróficos: Desenvolvimento e Desempenho.

RESUMO

O monitoramento *inline* do cultivo de micro-organismos autotróficos permite não só o acompanhamento da concentração de biomassa, mas a tomada de decisão sobre a operação de fotobioreatores, em especial quanto à decisão de políticas de operação. A disponibilidade de equipamentos de monitoramento de cultivo sugere a necessidade de avaliação da capacidade de processamento simultâneo de informações de monitoramento e também de modelamento cinético em tempo real. O uso do hardware livre Arduino e placas compatíveis representa um ferramental novo, de baixo custo e consumo de energia, dotado de entradas e saídas analógicas e digitais, útil tanto para a automação como para a simulação e otimização de processos. Por outro lado, a pouca memória, a possibilidade de erro de truncamento em operações de ponto flutuante e o clock baixo podem prejudicar o uso da plataforma Arduino em processamento numérico. Cabe ressaltar que algumas das versões de microcontroladores a quantidade de memória disponível é pequena, da ordem de 28 Kbytes livres e que o poder de processamento numérico é pequeno, quando comparado com outros tipos de processadores. Este trabalho teve por objetivo construir e avaliar quanto o desempenho uma aplicação computacional de modelamento cinético baseada nos modelo empírico através da equação de Verhulst, na plataforma Arduino, usando o método de minimização do erro quadrático baseado no algoritmo de otimização por enxame de partículas (PSO). A escolha pelo algoritmo PSO foi fundamentada pelo mesmo ser de uso geral, o que possibilitaria a fácil troca da equação do modelo empírico utilizado. Foi testada a mesma aplicação computacional nas três placas microcontroladas da família Arduino: Due, Mega e Nano. Devido as características diferentes, o melhor desempenho quanto ao tempo computacional foi o da placa Arduino Due, cerca de 4 vezes mais rápido, mas todas os três conjuntos embarcados foram capazes de modelar o conjunto de dados experimentais com a equação logística, em um tempo factível para aplicações *inline*, menor do que 2 min, e com coeficientes de determinação (R^2) 0,97, usando uma configuração para o algoritmo PSO de 10 partículas, convergindo em menos de 500 iterações. Apesar da frugalidade do hardware do microcontrolador de 8 bits, o mesmo foi capaz de suprir condições de processamento que o qualificariam para aplicações de modelagem empírica durante o acompanhamento de cultivo de microalgas.

Palavras chave: modelo empírico, modelo logístico, otimização por enxame de partículas

Autotrophic Bioprocess Kinetic Modeling through Arduino Microcontroller: Development and Performance

ABSTRACT

Inline monitoring of autotrophic cultures allows not only the visualization of instantaneous process data, but also the ability of generating commands based on such data. However, in order to correctly utilize the available information, it is necessary to develop a control system capable of dealing with the influx of kinetic data. The utilization of open hardware, such as the Arduino structure and other compatible boards, represents a new strand of hardware tools. This architecture possesses fully programmable digital and analogical inputs and outputs and low energy demand and price when compared to other proprietary hardwares. Generating a great interest in the automation community. However, the possibility of numerical errors, floating point arithmetics inconsistencies and low clock processing speed may pose as a threat to the utilization of this architecture in numerical processing. It is important to emphasize that some versions of the Arduino family of boards have small amounts of free memory, around 28 Kbytes, generating a small numerical processing capacity when compared to other microcontrollers. This work objective was to construct and evaluate an application that solves an kinetic empirical model based in the Verhulst equation inside the Arduino platform. The numerical optimization method utilized to minimize the sum of the squared errors was a Particle Swarm Optimization (PSO) algorithm. A PSO algorithm was utilized due to its high adaptability characteristics, enabling changes in the base kinetic model. The same algorithm was tested in three boards from the Arduino family: Due, Mega and Nano. Due to different characteristics, the best result was achieved by the Arduino Due board, arriving at the solution 4 times faster than the other boards. However, even in the other boards, the solution was reached in a time period small enough, under two minutes, to be applied in an inline situation. The coefficient of determination was similar in all the boards, around 0.97, when using 10 particles in the PSO swarm, converging after less than 500 iterations. Regarding the simplicity of the 8 bit architecture, the board with this architecture was able to comply with minimal demands for online empirical modeling, and this board may be used for microalgae culture monitoring, as well as the boards with more complex architectures.

Keywords: Empirical Modeling, Logistic Model , Particle Swarm Optimization,

1 INTRODUÇÃO

De forma semelhante ao advento da microcomputação nos anos 70 do século passado, a disponibilidade dos microcontroladores de baixo custo, de baixo consumo de energia, e com poder de processamento comparável a alguns microcomputadores de outrora, pode permitir hoje a simulação digital e o processamento numérico em dispositivos ainda menores, o que poderá servir como ferramenta na automação de processos fotoautotróficos e sistemas de cultivo de microalgas.

Por outro lado, as restrições de espaço de memória de programas e de dados, bem como a precisão numérica de operações de ponto flutuante, nessas plataformas microcontroladas, podem ser entraves à construção de aplicações com as mesmas na automação de bioprocessos.

Este trabalho teve por objetivo avaliar o desempenho computacional quanto à velocidade de processamento, quanto a precisão e quanto a exatidão dos resultados oriundos de implementações do ajuste de equações cinéticas usando o método da minimização do somatório do erro quadrático pelo método da otimização do enxame de partículas na plataforma microcontrolada Arduino.

O microcontrolador Arduino é um hardware de controle *open source* o qual é projetado para ser facilmente integrado aos sensores e atuadores (www.arduino.cc). Uma das mais importantes características é a facilidade de programação mesmo por usuários com pouca experiência técnica na programação de hardware embarcado (LEUNG,2013). A plataforma Arduino foi concebida em 2005 por Barragan, Banzi, Cuartielles, Mellis, Marino e Zambeti (GIBB, 2010), motivados pela necessidade de ensinar em pouco tempo eletrônica prática para estudantes de design do Instituto Ivrea (Banzi, 2011).

Recentemente a plataforma Arduino passou a ser usada para os mais diferentes fins, por pesquisadores dos mais diversos campos, tais como engenharia (CELA et al.,2013;ORTEGA-ZAMORANO et al.,2013;KORNUTA et al., 2013), como ferramenta em *datalogging* (ELMEHRAZ et al., 2013; PURVIS et al. 2013, BORKAR, 2012; HLOUPIS et al., 2012), e em automação de bioprocessos (FALK, 2011).

As aplicações citadas focam em monitoramento, automação, na construção de dispositivos *stand alone*, reais, não no uso do Arduino em simulação numérica ou no processamento de grandes quantidades de dados, ou envolvendo algoritmos iterativos. Um

dos motivos para isso é que enquanto normalmente constituinte de um sistema embarcado, o microcontrolador não apresenta uma interface de entrada/saída pronta. A mesma deve ser implementada especificamente para cada aplicação. De acordo com HEATH (2003) um sistema embarcado é um sistema baseado em microprocessador que é construído para controlar uma função dedicada, em um sistema físico maior.

A utilização de microcontroladores de pouco poder computacional é restrita na simulação digital e na modelagem cinética de bioprocessos, quando comparado com o uso de outras plataformas de hardware computacional.

Um aspecto interessante a ser considerado é o condizente com o consumo de energia durante o uso. Segundo Göddeke et al.(2013) o uso de soluções de computação baseadas em processadores de arquitetura ARM são mais eficientes energeticamente do que o uso para a solução do mesmo problema através da plataforma x86. Sob o ponto de vista do poder computacional, da flexibilidade dos sistemas computacionais multipropósito, e da interface com usuário, os sistemas computacionais x86 são melhores do que os microcontroladores, para a simulação. A grande vantagem do microcontrolador é o baixíssimo custo, e sua robustez.

Ocorre que na construção de dispositivos de campo, que monitorem ou acompanhem bioprocessos, em planta piloto ou real grandeza, a possibilidade de contar com ferramentas de simulação digital na ferramenta que permite a construção do sistema embarcado autônomo é vantajosa. Assim, além de contar com hardware de monitoramento que registra informações e as grava ou transmite para pós-processamento, o processamento desses dados pode ser feito localmente no microcontrolador e fornecer os parâmetros para outras aplicações, como covalidação, decisão sobre estratégias de operação, conciliação de informações de balanço material, controle, dentre outras possíveis.

A modelagem empírica de bioprocessos, chamada de modelagem “caixa preta” considera o sistema a ser modelado em termos das entradas e saídas características, consistindo de combinações de funções matemáticas os quais não permitem qualquer interpretação física, sendo mais usados quando não há informação a priori do sistema (GROSFILS, 2007).

Apesar de considerar somente uma relação entre entrada e saída de maneira empírica, não fazendo uso das equações dos princípios de conservação – da massa, energia e

quantidade de movimento - por exemplo, que permitem montar uma representação matemática que aumenta a validade do modelo para fora dos intervalos e faixas de entrada e saída testados nos modelos “caixa preta”, mas os mesmos apresentam a possibilidade de serem facilmente implementados em hardwares embarcados.

Especificamente no caso de implementação em um hardware autônomo de acompanhamento de cultivo de microalgas, a construção de uma ferramenta de modelagem empírica poderia servir para determinar não só parâmetros como velocidade específica máxima de crescimento, ou tempo de gerações para caracterizar o cultivo, mas também para prever a potencial produtividade máxima.

O uso da modelagem *black box* em um sistema embarcado autônomo para modelagem de bioprocessos necessita de um método amplo de determinação dos parâmetros do modelo matemático escolhido, como por exemplo a minimização do somatório do erro quadrático, através de um método heurístico.

Um exemplo de modelos empírico associado com o crescimento de micro-organismos, pode ser o modelo logístico de Verhulst:

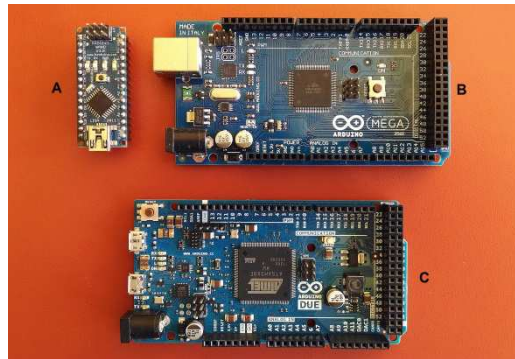
$$X(t) = k_1 / (k_2 + k_3 \cdot e^{-k_4 \cdot t}) \quad (1)$$

Onde $X(t)$ é a concentração de biomassa ao longo do tempo, e as constantes k_1 , k_2 , k_3 , k_4 as constantes do modelo a serem determinados pelo procedimento de ajuste cinético e t o tempo.

O parâmetro k_1 no modelo logístico de Verhulst está associado com a capacidade do sistema sustentar a população de micro-organismos, sendo portanto um parâmetro indicativo da concentração máxima de biomassa passível de ser alcançada no cultivo, em analogia com a modelagem de populações conforme Gatto e colaboradores (1988).

2 MATERIAIS E MÉTODOS

Foram usadas na realização das simulações para avaliação de desempenho as placas Arduinos Nano (Funduino, China), Mega (China) e Due (Arduino Store, Itália), as quais usam, respectivamente os microcontroladores ATmega328 e ATmega2560 (ambas 8 bits), e SAM3X8E Atmel ARM CORTEX – M3 de 32 bits. As placas são mostradas na Figura 1, observa-se as diferenças construtivas dos 3 modelos.

Figura 1 – (A) Arduino Nano, (B) Arduino Mega e (C) Arduino Due

As principais características de hardware encontram-se na Tabela 1. O Arduino Nano, o menor fisicamente das três configurações, também é aquele mais escasso em recursos de memória, sendo o de menor custo das três placas utilizadas.

O Arduino Mega, é na realidade uma ampliação do Arduino Uno, com maior número de portas seriais, entradas e saídas, e com 8 vezes mais memória do que o Arduino Nano.

Tabela 1 – Principais Características das Placas Arduinos Utilizadas

	Arduino Nano	Arduino Mega	Arduino Due
Microcontrolador	ATmega328	ATmega2560	AT91SAM3X8E
Velocidade Clock	16 MHz	16 MHz	84 MHz
Memória Flash	32 kByte (2 kBytes do <i>boot loader</i>)	256 kByte (8 kbyte do <i>boot loader</i>)	512 kbyte
SRAM	2 kbyte	8 kByte	96 kbyte
EEPROM	1 kbyte	4 kbyte	-
Custo (Brasil)	R\$ 40,00	R\$ 70,00	R\$ 120,00
Arquitetura	8 bit	8 bit	32 bit – Cortex M3

A placa mais recente entre as testadas, de 32 bits, além de mais memória apresenta o diferencial de ter uma velocidade de clock mais de 5 vezes maior que as placas de 8 bits.

A forma de interface utilizada foi através do uso da comunicação serial, pela porta USB de um notebook servindo simplesmente como dispositivo de E/S da saída formatada de resultados numéricos do programa desenvolvido nos microcontroladores Arduino.

Com o intuito de avaliar o desempenho do ajuste cinético para o modelo de Verhulst, ou modelo logístico, foram usados no teste do desempenho nas placas com microcontroladores os dados de cultivo de *Spirulina* sp. LEB 18 em fotobiorreator *raceway outdoor*. Segundo Arbib et al. (2013) o modelo logístico de Verhulst pode descrever o crescimento da biomassa de microalgas em diferentes condições comuns ao cultivo em biorreatores operando em batelada.

Para o conjunto de dados de concentração de biomassa, em função do tempo, para 3 corridas experimentais de cultivo de *Spirulina* sp. LEB 18, fez-se ajustes usando o modelo logístico de Verhulst em uma implementação que usa o algoritmo da otimização do enxame de partículas, com o intuito de minimizar o somatório do erro quadrático entre a concentração determinada pelo modelo e a concentração de *Spirulina* medida durante cultivo. O algoritmo PSO permite a determinação das constantes do modelo.

3 RESULTADOS E DISCUSSÃO

O mesmo programa de modelagem empírica, implementado e testado nas 3 versões de placas Arduino apresentou os tempos de execução mostrados na Tabela 1, para o mesmo numero total de iterações, 1000 e com os mesmos parâmetros de ajuste do algoritmo de otimização. Conforme esperado a plataforma de 32 bits, o Arduino Due, foi bastante mais eficiente em termos computacionais, sendo de 3 a 4 vezes mais rápido do que as outras 2 placas testadas.

Quando do uso do Arduino Nano para a modelagem, houve a necessidade de limitar o teste a no máximo 10 partículas, devido à sua limitada memória para dados, que na implementação usada era esgotada com máximo 12 partículas.

Os tempos próximos entre o Arduino Nano e o Arduino Mega estão vinculados ao fato dos mesmos usarem o mesmo microcontrolador ATmega328. A pequena diferença está associada às diferenças inerentes ao circuito eletrônico da comunicação serial das 2 placas. Cabe ressaltar que o uso da comunicação serial, durante as corridas experimentais realizadas, implica em um tempo computacional importante, que não seria dispendido em uma aplicação de campo.

Tabela 2 - Tempos de Execução do Programa de Modelagem Cinética

	Tempo (s) em função do número de partículas (p)				
	5 p	10 p.	15 p.	20 p.	25 p.
Due	10,5	18,9	27,4	35,8	44,6
Mega	35,1	72,3	106,2	138,6	179,2
Nano	34,1	70,2	-	-	-

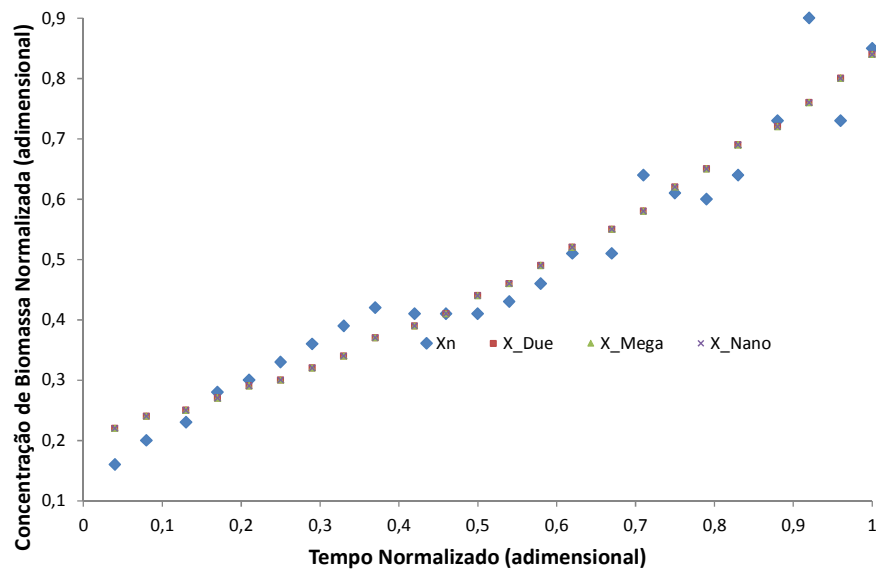
Quando foi avaliado o efeito do número de partículas no ajuste, conforme mostra a Tabela 2, a partir de 10 partículas o ajuste obtido foi o mesmo, o que justificaria o uso de tal quantidade de partículas na implementação do programa nas três plataformas testadas, já que para maior número de partículas usadas há dispêndio de memória, sem ganho na ajuste e com perda de desempenho, pois há maior

A limitação de memória de dados da placa Arduino Nano pode ser contornada, com um reprojeto do programa de modelagem, visando otimizar não só o desempenho do ajuste, mas a ocupação de memória, com o uso de alocação dinâmica.

Tabela 3 - Coeficiente de Determinação, R^2 , do Modelo Cinético Logístico de Verhulst

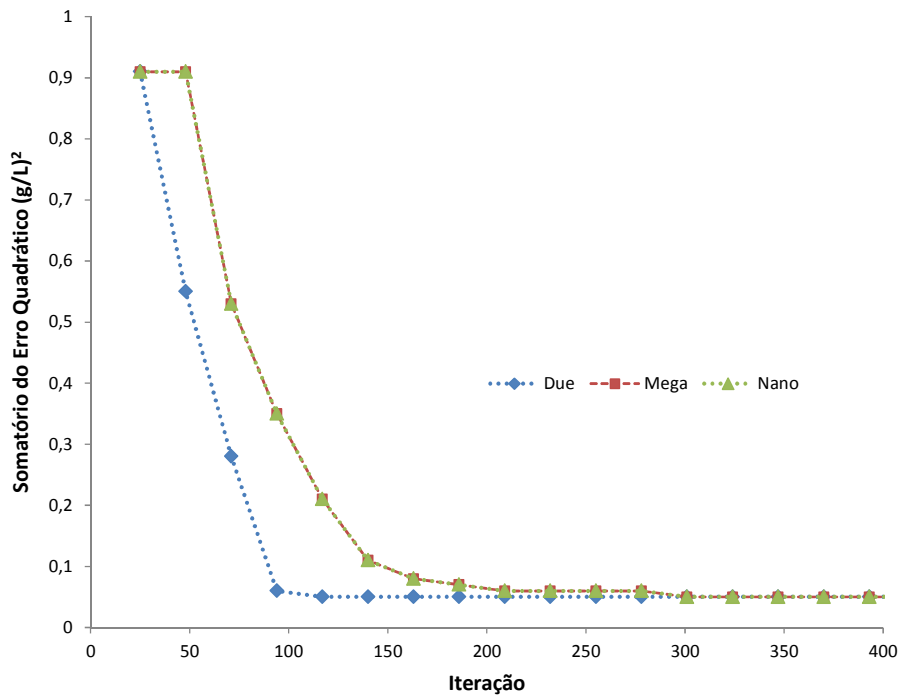
Coeficiente de Determinação, R^2 , em função do número de partículas (p)					
	5 p.	10 p.	15 p.	20 p.	25 p.
Due	0,92	0,97	0,97	0,97	0,97
Mega	0,96	0,97	0,97	0,97	0,97
Nano	0,96	0,97	-	-	-

A qualidade do ajuste cinético obtido é mostrada na Figura 2, onde se observa que as 3 placas testadas alcançaram no modelamento resultado equivalente. A diferença foi no número de iterações necessárias para alcançar o mesmo somatório do erro quadrático, conforme a Figura 3. Novamente, a plataforma de 32 bits apresentou desempenho melhor, com a convergência mais rápida na determinação dos parâmetros do modelo cinético de Verhulst.

Figura 2 – Concentrações de Biomassa – Experimental (X_n) e Determinadas pelo Ajuste dos Modelos nas Placas Arduino (X_{Due} , X_{Mega} , X_{Nano})

A partir da análise conjunta das Figuras 2 e 3 foi possível inferir que para o tipo de equação e para o algoritmo testado um número de 400 iterações foi suficiente para se alcançar a representatividade desejável na modelagem.

Figura 3 – Minimização do Somatório do Erro Quadrático ao Longo das Iterações no Ajuste dos Modelos nas Placas Arduino



Dentre os parâmetros do algoritmo PSO o de maior impacto em uma implementação em microcontrolador é o tamanho da população, pois o aumento do mesmo implica necessariamente em um aumento muito significativo da memória ocupada para variáveis locais do programa, pois essa quantidade é diretamente proporcional à duas vezes o tamanho da população vezes o número de parâmetros do modelo. Para um modelo de 4 parâmetros, como o logístico, dobrar o tamanho da população representa em aumentar em 8 vezes a memória alocada.

Dependendo da quantidade de memória alocada inicialmente essa implementação é inviável no Arduino Nano, pois o mesmo é baseado no ATmega328 o que implica que somente 2048 bytes de memória para variáveis.

Outra limitação importante é quanto ao tempo de execução. Para aplicações *inline*, é desejável que o tempo total não seja maior do que 3 min. Esse limite é pensado para uso em

aplicações em que seria feito o *datalogging* com o mesmo equipamento, e um tempo de amostragem de 5 minutos, não ficaria de forma alguma prejudicado por esse limite de tempo.

A maior limitação do uso do Arduino Nano, combinando função de *datalogging* e de modelagem cinética no algoritmo desenvolvido seria em termos de memória para programa.

Da experiência de desenvolvimento anterior, o espaço de somente 14 kbytes para o programa de *datalogging* seria exíguo, o que exigiria um esforço de desenvolvimento para reduzir o tamanho ocupado pelo código.

4 CONCLUSÕES

O uso de placas da plataforma Arduino fomenta a prototipagem rápida de sistemas embarcados para aplicação em monitoramento, controle, modelagem e simulação de bioprocessos, devido a facilidade de uso, pequeno consumo de energia, custo extremamente baixo, disponibilidade de portas analógicas de aquisição de sinal, de saídas digitais, inclusive algumas com a possibilidade de usarem a modulação por largura de pulso para emular uma saída analógica.

No campo da modelagem empírica da cinética de cultivo de microalgas, apesar do pequeno poder computacional, o uso do Arduino é uma real possibilidade, pois o programa desenvolvido neste trabalho foi capaz de ajustar uma equação empírica do tipo logística com um coeficiente de determinação (R^2) 0,97, para as 3 arquiteturas testadas.

O tempo computacional global, nas corridas realizadas foi compatível com uma possível implementação em um hardware de monitoramento de campo para o acompanhamento *in line* da concentração de biomassa, pois mesmo com o número de iterações superdimensionado em 1000 iterações, e com tamanho de população do algoritmo PSO de 10 partículas o mesmo foi 70,2 s para a placa Arduino Nano.

A viabilidade de projeto e construção de ferramentas de software úteis para a modelagem cinética implementados em microcontroladores, visando a simultânea tarefa de adquirir dados de cultivo, monitorando o mesmo e já processar as informações de concentração de biomassa microalgal, a medida que a mesmas estiverem sendo obtidas, surgiu seria um avanço na inteligência do processo de cultivo de microalgas.

5 REFERÊNCIAS

ARBIB, Z.; RUIZ, J.; Álvarez-Dias, P.; GARRIDO-Pérez, G., PERALES, J. A.; Capability of different microalgae species for phytoremediation processes: Wastewater tertiary treatment, CO₂ bio-fixation and low cost biofuels production, *Water Research*, 2013.

BANZI, M.; *Getting Started with Arduino*, O'Reilly Media, 2009.

BORKAR, C. & XINRONG, L. ; Development of wireless sensor network system for indoor air quality monitoring, North Texas University, 2012,

CELA, A.; YEBES, J.J.; ARROYO, R.; BERGASA, L.M.; BAREA, R.; LÓPEZ, E.; Complete low-cost implementation of a teleoperated control system for a humanoid robot. *Sensors* v.13, 2013.

ELMEHRAZ, N.; Design of a Highly Portable Data Logging Embedded System for Naturalistic Motorcycle Study, University of South Florida, 2013.

FALK, H.M.; BENZ, H.C.; Monitoring the anaerobic digestion process, Jacobs University, 2011.

GATTO, M.; MURATORI, S.; RINALDI, S.; A Functional Interpretation of the Logistic Equation. *Ecological Modelling*, v.42, 2, pp. 155-159, 1988.

GIBB, A. M., *New Media Art, Design, And The Arduino Microcontroller: A Malleable Pool*, Master's Thesis, Pratt Institute, 2010.

GROSFILS, A.; First principles and black box modelling of biological systems; Doctorat en Sciences Appliquées, Université Libre de Bruxelles, 2007. <http://theses.ulb.ac.be/ETD-db/collection/available/ULBetd-09122007-105503>

GÖDDEKE, D.; KOMATITSCH, D. ; GEVELER, M.; RIBBROCK, D.; RAJOVIC, N.; PUZOVIC, N.; RAMIREZ, A.; Energy efficiency vs. performance of the numerical solution of PDEs: An application study on a low-power ARM-based cluster, *Journal of Computational Physics*, Vol. 237, pp 132-150, 2013.

HLOUPIS, G.; STAVRAKAS, I.; MOUTZOURIS, K.; ALEXANDRIDIS, A.; TRIANTIS, D.; WSN Open Source Development Platform: Application to Green Learning, *Procedia Engineering* 25, 1049 – 1052, 2011.

KHAOUANE, L.; BENKORTBI, O.; HANINI, S. and SI-MOUSSA, C.. Modeling of an industrial process of pleuromutilin fermentation using feed-forward neural networks. *Brazilian Journal of Chemical Engineering*, vol.30, n.1, pp. 105-116, 2013.

KORNUTA, J.A., NIPPER, M.E., DIXON, B. , Low-cost microcontroller platform for studying lymphatic biomechanics in vitro. *J. Biomech.* 46(1), 183–186, 2013.

LEUNG, K. - A History of the Arduino Microcontroller – disponível em <http://www.kenleung.ca/portfolio/arduino-a-brief-history-3/>. Acessado em 10/04/2014.

ORTEGA-ZAMORANO, F., SUBIRATS, J.L., JEREZ, J.M., MOLINA, I., FRANCO, L., Implementation of the C-Mantec neural network constructive algorithm in an arduino UNO microcontroller. *Lect. Notes Comput. Sci.* 7902, 80–87, 2013.

PURVIS, M.; REMOTE DATA LOGGER For Photo-Voltaic Research, Murdoch University, 2013.

VICENTE, M. J. P.; SARMENTO, M. H. C.; DECT Shield for Arduino, IFT Lisboa, 2012.

CAPÍTULO IV: CONCLUSÕES GERAIS

6 CONCLUSÕES GERAIS

A partir do uso de um hardware baseado em microcontrolador da plataforma Arduino, foram construídos protótipos de dispositivos de monitoramento em fotobiorreator *raceway*, para aquisição, registro e tratamento das variáveis do processo de cultivo de *Spirulina*: temperatura, iluminância, concentração da suspensão de biomassa e pH.

Os protótipos construídos foram testados ao longo de corridas experimentais em estufa *outdoor*, suportando as condições de temperatura e umidade ambiente, que variaram entre 18° C até mais de 60° C no interior da estufa de cultivo.

Foi possível determinar o padrão da relação entre a iluminância na suspensão de *Spirulina* durante o cultivo e fora do bioreator, bem como registrar a evolução do crescimento, pela queda da intensidade do sinal no receptor usado no sistema óptico desenvolvido.

Foi constatado que para o fotobiorreator usado, há uma razão máxima entre iluminâncias interna e externa variável entre 47% até 42% durante o cultivo de *Spirulina* sp. LEB 18.

As medidas realizadas no sensor óptico baseado na medida da intensidade da radiação monocromática passante através da suspensão, usado em condições *outdoor* somente teve correlação entre a concentração de biomassa e a radiação para concentrações de suspensão de cianobactéria abaixo de 0,6 g/L.

A partir da identificação da necessidade de avaliar outro princípio físico de transmissão da luz no sistema óptico, foi avaliada a relação entre o espalhamento de radiação monocromática e a concentração de biomassa de *Spirulina*, em laboratório.

Houve correlação estatística ($R^2 > 0,95$) entre o sinal detectado pelo sensor TAOS TSL235R e a concentração de biomassa de *Spirulina*, durante o espalhamento da radiação de 530 nm, especificamente para os ângulos entre emissor e receptor de 45°, 90° e 135°.

As medições de espalhamento da radiação monocromática em 530 nm pela suspensão de *Spirulina* que tiveram maior sensibilidade, na faixa entre 0,2 g/L e 1,4 g/L foram aquelas com um ângulo entre emissor e receptor de 90°, sendo 9,1% em média maiores do que com o ângulo de 45°.

O uso de medidas de espalhamento de radiação luminosa monocromática em 530 nm se mostrou capaz de propiciar fundamentação para o treinamento de um sensor baseado em software, usando uma arquitetura *neurofuzzy* do tipo Takagi-Sugeno-Kang, com o uso do

algoritmo de otimização pelo enxame de partículas, com coeficiente de determinação (R^2) 0,99 usando informações de espalhamento da radiação monocromática de 530 nm, em suspensão de *Spirulina*, obtidas com ângulos de 45° e 135° entre emissor e receptor. A partir dessa constatação, foi concebida uma proposta de protótipo de sensor combinado.

Foi também avaliada a viabilidade de efetuar a modelagem cinética empírica através do modelo logístico, no mesmo hardware desenvolvido, baseado em microcontrolador. A aplicação implementada no microcontrolador foi capaz de ajustar uma equação empírica do tipo logística para a cinética de cultivo da microalga *Spirulina* com um coeficiente de determinação (R^2) 0,97, para as 3 arquiteturas diferentes de microcontroladores testados.

O tempo computacional global, nas corridas realizadas foi compatível com uma possível implementação em um hardware autônomo de monitoramento de campo para o acompanhamento *in line* da concentração de biomassa, pois mesmo com o número de iterações superdimensionado em 1000 iterações, e com tamanho de população do algoritmo PSO de 10 partículas o mesmo foi 70,2 s para a placa Arduino Nano, a de menor poder computacional entre as 3 testadas.

Apesar da relativa simplicidade do hardware do microcontrolador de 8 bits, o mesmo foi capaz de suprir condições de processamento que o qualificariam para aplicações de modelagem empírica durante o acompanhamento de cultivo de microalgas.

7 SUGESTÕES PARA TRABALHOS FUTUROS

Realizar estudos experimentais de cultivo de *Spirulina* sp LEB 18 no fotobiorreator de escala piloto, utilizando sensores de dióxido de carbono dissolvido, de oxigênio dissolvido e de pH para acompanhar o cultivo de microalgas, juntamente com o protótipo MARK que foi desenvolvido neste trabalho, usando a comunicação de 4 a 20 mA, com resistor de 220 Ω para gerar o sinal de queda de tensão para o sistema Arduino realizar a aquisição através de porta analógica, efetivando a leitura em tensão.

Desenvolver a automação no cultivo de microalgas em reator *raceway*, fechando a malha de controle, usando como variável controlada a iluminância no meio de cultivo, e manipulando a taxa de diluição do meio, através da retirada de cultivo e/ou do acréscimo de meio de cultivo ou solvente, na operação semi-contínua do biorreator.

Testar em condições operacionais *outdoor* o sensor óptico combinado de biomassa proposto neste trabalho, não só quanto às figuras de mérito analítico, mas também quanto às dificuldades associadas à agregação de mucopolissacarídeos na unidade de recepção e de emissão do laser, e com outros micro-organismos fotoautotróficos de morfologia diferenciada.

Construir o modelo matemático do balanço material associado com a biofixação de dióxido de carbono e implementá-lo para o acompanhamento do processo com o processamento numérico no microcontrolador do protótipo MARK.

Adaptar o sistema ANFIS usado nesse trabalho para a automação e controle da taxa de diluição, da vazão de dióxido de carbono acrescido ao cultivo e da concentração de oxigênio dissolvido na operação de fotobiorreatores.

Avaliar o espalhamento da radiação quanto a outros ângulos entre emissor e receptor de radiação monocromática.

Construir o sistema supervisor para o controle da produção de biomassa, usufruindo da facilidade de comunicação sem fio, através de redes *bluetooth* e/ou *wifi* com o protótipo MARK, de forma a tornar as informações de cultivo acessíveis via INTERNET, em tempo real.

Investigar o uso do sistema óptico de determinação de concentração de biomassa baseado no espalhamento da radiação luminosa monocromática para outros bioprocessos envolvendo outros microorganismos.

CAPÍTULO V: REFERÊNCIAS

REFERÊNCIAS

- ACOSTA, R.; HIPPEL, E.; Open Source Hardware, Master's Thesis, MIT, 2009.
- AHSAN, M.; HABIB, M.; PARVIN, M.; HUNTINGTON, T.C.; HASAN, M. R.; A Review on Culture, Production and Use of *Spirulina* as Food for Humans and Feeds for Domestic Animals and Fish; Food and Agriculture Organization of the United Nations, FAO, Rome, 2008.
- ANDRADE, M. da R.; CAMERINI, F. V.; COSTA, J. A. V. Chemical carbon losses and growth kinetics in *Spirulina* cultures. *Química Nova*, v. 31, n. 8, p. 2031–2034, jan. 2008.
- ANDRADE, M.R. ; RADMANN, E.M. ; MORAIS, M.G.; TEIXEIRA, G.G. ; BRUSCH, L.R.F. ; COSTA, J.A.V. ; Pilot Scale Semicontinuous Production of *Spirulina* Biomass in Southern Brazil. *Aquaculture (Amsterdam)*, v. 294, p. 60-64, 2009.
- ARBIB, Z.; RUIZ, J.; Álvarez-Dias, P.; GARRIDO-Pérez, G., PERALES, J. A.; Capability of different microalgae species for phytoremediation processes: Wastewater tertiary treatment, CO₂ bio-fixation and low cost biofuels production, *Water Research*, 2013.
- ATALA, D. I. P. ; LINDE, G. A. ; MIBIELLI, G. M. ; ARRIECHE, L. ; KRÜGER, R. T. ; COSTA, J. A. V. . Estudo do Cultivo da Cianobactéria *Spirulina Platensis* em Incubador Rotativo (Shaker). In: *Anais da VII Mostra de Iniciação Científica*, v. 1. p. 33., Passo Fundo, 1997.
- BANZI, M.; *Getting Started with Arduino*, O'Reilly Media, 2009.
- BANZI, M.; Interview – disponível em <http://arstechnica.com/information-technology/2013/10/arduino-creator-explains-why-open-source-matters-in-hardware-too>, acessada em 20/07/2014.
- BARANYI, J.; ROBERTS, T.A., A Dynamic Approach to Predicting Bacterial Growth in Food. *Int. J. Food Microbiol.* 23. 277-29, 1994.
- BARROS, K. K. S., Produção de biomassa de *Arthrospira platensis* (*Spirulina platensis*) para alimentação humana. Dissertação (Mestrado em Ciência e Tecnologia de Alimentos) - Universidade Federal da Paraíba, 2010.
- BAUGHMAN, D. R., LIU, Y. A., *Neural Networks in Bioprocessing and Chemical Engineering*. 1st Ed. , Academic Press Inc., 487 p., 1995.
- BERGMANN, N.W.; WALLACE, M.; CALIA, E.; Low cost prototyping system for sensor networks, 6th International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Brisbane, pp. 19-24, 2010.

BEUERMANN, T.; EGLY, D.; GEOERG, D.; KLUG, K. I.; STORHAS, W.; METHNER, F.J.; On-line carbon balance of yeast fermentations using miniaturized optical sensors. PMID: 22100900: Journal of Bioscience and Bioengineering, mar. 2012. v. 113, n. 3, p. 399–405.

BIASIZZO, K. K., SKRJANC, I., MATKO, D., Fuzzy Predictive Control of Highly Nonlinear pH Process. Computers Chem. Engng., Vol 21, Suppl., pp. S613-S618, 1997.

BOHREN, C.F.; HUFFMAN, D.R.; Absorption and Scattering of Light by Small Particles; John Wiley, New York, 1983.

BONISSONE, P.P.; Adaptive neural Fuzzy Inference Systems (ANFIS) Analysis And Applications, 1997. In: <http://homepages.rpi.edu/~bonisp/fuzzy-course/99/L9/ANFIS.pdf>

BORKAR, C. & XINRONG, L. ; Development of wireless sensor network system for indoor air quality monitoring, North Texas University, 2012,

BUTTERFIELD, A.; Use of Arduino Microcontrollers in Chemical Engineering Curricula, Proceedings of AIChE Annual Meeting, 2013.

CAMERINI, F. V. Fixação biológica de dióxido de carbono por *Spirulina (Arthrospira) platensis*. Dissertação de Mestrado em Engenharia e Ciência de Alimentos – FURG, Rio Grande, 2008.

CARVALHO, L. F. Desenvolvimento de novos alimentos para praticantes de atividade física adicionados ou não de *Spirulina*. Dissertação de Mestrado em Engenharia e Ciência de Alimentos – FURG, Rio Grande, 2010.

CARVALHO, L.F. de; Produção de biossurfactantes e nanoemulsões a partir da microalga *Spirulina*. Tese de Doutorado em Engenharia e Ciência de Alimentos – FURG, Rio Grande, 2014.

CASTELIJN, A.A.; BENTHEM, R.C.; ASSEM, D.; Self compensating real-time biomass sensor. Proceedings of the First International Symposium on Microgravity Research & Applications in Physical Sciences & Biotechnology, Sorrento, Italy, 2000.

CELA, A.; YEBES, J.J.; ARROYO, R.; BERGASA, L.M.; BAREA, R.; LÓPEZ, E.; Complete low-cost implementation of a teleoperated control system for a humanoid robot. Sensors v.13, 2013.

CHISTI, Y. Biodiesel from Microalgae. Biotechnology Advances, 25, 294-306, 2007.

CHRISTENSON, L.; SIMS, L., Production and Harvesting of Microalgae for Wastewater Treatment, Biofuels, and Bioproducts. Biotechnology Advances 29, 686–702, 2011.

COGNE, G., LASSEUR, C.H., CORNET, J.F., DUSSAP, C.G., GROS, J.B.; Growth monitoring of a photosynthetic micro-organism (*Spirulina platensis*) by pressure measurements. Biotechnol. Lett. 23, 1309–1314, 2001

COLLA, L. M. ; DUARTE FILHO, P. ; COSTA, J. A. V. ;. *Spirulina Platensis* Growth in Open Raceway Ponds using Fresh Water Supplemented With Carbon, Nitrogen and Metal Ions. *Zeitschrift für Naturforschung. C, A Journal of Biosciences*, v. 58c, p. 76-80, 2003.

COLLA, L. M. ; BERTOLIN, T. E., COSTA, J. A. V. ; Fatty Acids Profile of *Spirulina platensis* Grown Under Different Temperatures and Nitrogen Concentrations, *Zeitschrift für Naturforschung. C, A Journal of Biosciences*, v. 59, n. c, p. 55-59, 2004.

COSTA, J.A.V.; COLLA, L.M.; DUARTE FILHO, P.; KABKE, K.; WEBER, A. Modeling of *Spirulina platensis* growth in fresh water using response surface methodology. *World J. Microb. Biot.*, v. 18, p. 603-607, 2002.

COSTA, J. A. V.; MORAIS, M. G., The Role of Biochemical Engineering in the Production of Biofuels from Microalgae. *Bioresource Technology*, 102, 2–9, 2011.

COX, R. P.; MILLER, M.; NIELSEN, J.B.; NIELSEN, M.; THONSON, J. K.; Continous turbidometric measurements of microbial cell density in bioreactors using a light emitting diode and a photodiode. *Journal of Microbiological Methods* 10, 25-31, 1989.

COZZA, K. L. ; OLIVEIRA, L. ; MAGAGNIN, G.; COSTA, J. A. V. Different nitrogen sources and growth responses of *Spirulina platensis* in microenvironments. *World Journal of Microbiology & Biotechnology*, v. 17, n. 5, p. 439-442, 2001.

DASTIDAR, S. G.; Gompertz: A Scilab Program for Estimating Gompertz Curve Using Gauss-Newton Method of Least Squares, *Journal of Statistical Software*, 15,12 2006.

DRIANKOV, D., HELLENDORF, H., REINFRANK, M., An Introduction to Fuzzy Control. 2. Ed. , Springer-Verlag, New York, 316 p, 1996.

DUTRA PEREIRA, R.D., PEREIRA, J.A., FRATINI, A.M.F.; Desenvolvimento e Implementação de Controladores Nebulosos em uma Coluna Piloto de Destilação em Batelada. Dissertação de Mestrado em Sistema de Processos Químicos e Informática – Faculdade de Engenharia Química, Universidade Estadual de Campinas, Campinas, 1999.

EBERHART, R.C.; SHI, Y.; Particle Swarm Optimization: Developments, applications and Resources, *Proc. Congress on Evolutionary Computation*, IEEE, 2001.

ERDIRENCELEBI, D., YALPIR, S.; Adaptive network fuzzy inference system modeling for the input selection and prediction of anaerobic digestion effluent quality, *Applied Mathematical Modelling*, v.35, 8, pp 3821–3832, 2011.

ELMEHRAZ, N.; Design of a Highly Portable Data Logging Embedded System for Naturalistic Motorcycle Study, University of South Florida, 2013;

FALK, H.M.; BENZ, H.C.; Monitoring the anaerobic digestion process, Jacobs University, 2011.

FERREIRA, S.P.; Produção de lipídeos pela microalga *Chlorella* e obtenção de nanoemulsão de origem microalgal. Tese de Doutorado em Engenharia e Ciência de Alimentos – FURG, Rio Grande, 2013.

FURLONG, V.; DUTRA PEREIRA, R.; MARGARITES, A.C.; GOULART, P. ; COSTA, J.A.C. Modelagem Neuro-Fuzzy da Cinética de Crescimento da Microalga *Synechococcus nidulans*, Anais do IX SLACA, 2011.

FISHER, D. K.; GOULD, P.J.; Open-Source Hardware Is a Low-Cost Alternative for Scientific Instrumentation and Research, *Modern Instrumentation*, vol. 1, p.8-20, 2012.

GADKAR, K.G.; MEHRAB, S.; GOMES, J.; On-line adaptation of neural networks for bioprocess control, *Computers & Chemical Engineering*, Volume 29, Issue 5, 15 Pages 1047–1057, 2005.

GATTO, M.; MURATORI, S.; RINALDI, S.; A Functional Interpretation of the Logistic Equation. *Ecological Modelling*, v.42, 2, pp. 155-159, 1988.

GELB, A.; *Applied Optimal Estimation*, The MIT Press, 1974.

GERTZ, E.; JUSTO, P.D.; *Atmospheric Monitoring with Arduino: Building Simple Devices to Collect Data about the Environment*, O'Reilly Media, 2012.

GIBB, A. M., *New Media Art, Design, And The Arduino Microcontroller: A Malleable Pool*, Master's Thesis, Pratt Institute, 2010.

GITELSON, A. A. *et al.* Optical Properties of Dense Algal Cultures Outdoors and Their Application to Remote Estimation of Biomass and Pigment Concentration in *Spirulina Platensis* (cyanobacteria)1. *Journal of Phycology*, 1995. v. 31, n. 5, p. 828–834. . Acesso em: 11 fev. 2014.

GITELSON, A. A. *et al.* Optical Properties of Dense Algal Cultures Outdoors and Their Application to Remote Estimation of Biomass and Pigment Concentration in *Spirulina Platensis* (cyanobacteria)1. *Journal of Phycology*, v. 31, n. 5, p. 828–834, 1995.

GITELSON, A.; QIANG, H.; RICHMOND, A. Photic Volume in Photobioreactors Supporting Ultrahigh Population Densities of the Photoautotroph *Spirulina platensis*. PMID: 16535309 PMCID: PMC1388847: *Applied and Environmental Microbiology*, v. 62, n. 5, p. 1570–1573. 1996.

GÖDDEKE, D.; KOMATITSCH, D. ; GEVELER, M.; RIBBROCK, D.; RAJOVIC, N.; PUZOVIC, N.; RAMIREZ, A.; Energy efficiency vs. performance of the numerical solution of PDEs: An application study on a low-power ARM-based cluster, *Journal of Computational Physics*, Vol. 237, pp 132-150, 2013.

GODJEVAC, J. Comparison Between PID and Fuzzy Control. Internal Report R95.361. Ecole Polytechnique Fédérale, Lausanne, 1995.

GOMIDE, F. A. C., GUDWIN, R. R., Modelagem, Controle, Sistemas e Lógica Fuzzy. 1º Simpósio Brasil-Japão de Lógica Nebulosa. UNICAMP, Campinas, 1992.

GOMIDE, F. A. C., PEDRICZ, W., An Introduction to Fuzzy Sets – Analysis and Design. 1st Ed., MIT Press, 465 p., 1998.

GRIFFITHS, M. J. *et al.* Interference by pigment in the estimation of microalgal biomass concentration by optical density. Journal of microbiological methods, v. 85, n. 2, p. 119–123, 2011.

GROSFILS, A.; First principles and black box modelling of biological systems; Doctorat en Sciences Appliquées, Université Libre de Bruxelles, 2007, disponível em <http://theses.ulb.ac.be/ETD-db/collection/available/ULBetd-09122007-105503/>

HAYKIN, S.; Neural Networks – A Comprehensive Foundation, Macmillan Publishing Company, USA, 1994.

HAVLIK, I.; LINDNER, P.; SCHEPER, T.; REARDON, K.F.; On-line monitoring of large cultivations of microalgae and cyanobacteria, Trends in Biotechnology, Vol. 31, No. 7, July 2013.

HENRARD, A.S.A.; Produção e purificação de biogás utilizando microalga *Spirulina* sp. LEB-18. 2013. Tese de Doutorado em Engenharia e Ciência de Alimentos – FURG, Rio Grande, 2013.

HLOUPIS, G.; STAVRAKAS, I.; MOUTZOURIS, K.; ALEXANDRIDIS, A.; TRIANTIS, D.; WSN Open Source Development Platform: Application to Green Learning, Procedia Engineering 25, 1049 – 1052, 2011.

HUANG, Y.; LAN, Y.; THOMSON, S.J.; FANG, A.; HOFFMAN, W.C.; LACEY, R. E.; Development of soft computing and applications in agricultural and biological engineering, Computers and Electronics in Agriculture 71, 107–127, 2010.

JAGER, R., Fuzzy Logic in Control. Thesis Technische, Universiteit Delft, 1995.

KAPLAN, A.; REINHOLD, L.; CO₂ Concentrating Mechanisms in Photosynthetic Microorganisms. Annual Review of Plant Physiology and Plant Molecular, 50, 539-570, 1999.

KARR, C., GENTRY, E., Fuzzy Control of pH Using Genetic Algorithms. IEEE

Transactions on Fuzzy Systems, v. 1, 1, Feb. 1993, pp 46-53.

KENNEDY J., EBERHART, R.C.; Particle Swarm Optimization. Proc. IEEE Internacional Conference on Neural Networks, IEEE Service Center, Piscataway, NJ, IV: 1942-1948, Perth, Australia, 1995.

KENNEDY, J.; EBERHART, R.C.; Swarm Intelligence. Morgan Kaufmann. ISBN 1-55860-595-9, 2001.

KHAOUANE, L.; BENKORTBI, O.; HANINI, S. and SI-MOUSSA, C.. Modeling of an industrial process of pleuromutilin fermentation using feed-forward neural networks. *Brazilian Journal of Chemical Engineering*, vol.30, n.1, pp. 105-116, 2013.

KORNUTA, J.A., NIPPER, M.E., DIXON, B. , Low-cost microcontroller platform for studying lymphatic biomechanics in vitro. *J.Biomech.*46(1), 183–186, 2013.

LEUNG, K. - A History of the Arduino Microcontroller – disponível em <http://www.kenleung.ca/portfolio/arduino-a-brief-history-3/>. Acessado em 10/04/2014.

LINDE, G. A. ; ATALA, D. I. P. ; MIBIELLI, G. M. ; ARRIECHE, L. ; KRÜGER, R. T. COSTA, J. A. V.; Modelling of Growth Conditions for Cyanobacterium *Spirulina Platensis* in Microcosms. *World Journal of Microbiology & Biotechnology*, v. 16, p. 15-18, 2000.

LISBOA, C.R.; Produção de nanopartículas contendo peptídeos bioativos de microalgas. Tese de Doutorado em Engenharia e Ciência de Alimentos – FURG, Rio Grande, 2013.

LORENA, V.; SOUZA, A.M.; Análise Multivariada da Teoria à Prática, UFSM, CCNE, 2005

LU, Y.; *Industrial Intelligent Control – Fundamentals and Applications*, John Wiley and Sons, 1996.

MAMDANI, E. H., Applications of Fuzzy Algorithm for Simple Dynamic Plant. *Proceedings IEEE*, 121(12), pp. 1585-1588, 1974.

MAMDANI, E. H., ASSILIAN, S., An Experiment in Linguistic Synthesis With a Fuzzy Logic Controller. *International Journal of Man-Machine Studies*, 7, pp. 1-13, 1975.

MARGARITES, A. C. F. Seleção e cultivo de microalgas para produção de bioetanol. Dissertação de Mestrado em Engenharia e Ciência de Alimentos – FURG, Rio Grande, 2010.

MARGARITES, A.C.; Síntese de carboidratos por microalgas e produção de bioetanol. Tese de Doutorado em Engenharia e Ciência de Alimentos – FURG, Rio Grande, 2014.

MAROSE, S.; LINDEMANN, C.; ULBER, R.; SHEPER, T.; Optical sensor systems for bioprocess monitoring. *Trends in Biotechnology*, v. 17, n. 1, p. 30–34, 1999.

MARTINS, R.M.; Síntese, extração e purificação de biopolímeros extraídos de microalgas para desenvolvimento de nanofibras. Tese de Doutorado em Engenharia e Ciência de Alimentos – FURG, Rio Grande, 2014.

MEIRELES, L. A. *et al.* On-line control of light intensity in a microalgal bioreactor using a novel automatic system. *Enzyme and Microbial Technology*, v. 42, n. 7, p. 554–559, 2008.

MERRIAM-WEBSTER, Dicionário Online, [http:// m-w.com](http://m-w.com), acesso em junho de 2014.

MIE, G. von; Beiträge zur Optik trüber Medien, speziell kolloidaller Metallösungen. *Annalen der Physik*, n° 4, vol 25, p. 377-445, 1908.

MISHCHENKO, M. I.; TRAVIS, L. D.; LACIS, A.A.; Scattering, Absorption, and Emission of Light by Small Particles. Cambridge University Press, 2002.

MOLINA GRIMA, E.; ACIÉN FERNÁNDEZ, F.G.; GARCÍA CAMACHO, F., CHISTI, Y.; Photobioreactors: light regime, mass transfer, and scaleup. *Journal of Biotechnology* 70, 231-247, 1999.

MONK, S.; Programming Arduino: Getting Started with Sketches, Tab Books, 2011.

MORAIS, M. G. ; COSTA, J. A. V. ; MIRANDA, M. Z.; Biscoitos de Chocolate Enriquecidos com *Spirulina platensis*: Características Físico-Química, Sensorial e Digestibilidade. *Alimentos e Nutrição (UNESP)*, v. 17, p. 333-340, 2006.

MORAIS, M. G. ; COSTA, J. A. V.; Isolation and Selection of Microalgae from Coal Fired Thermoelectric Power for Biofixation of Carbon Dioxide. *Energy Conversion and Management*, v. 48, p. 2169-2173, 2007.

MORAIS, M. G. Bioengenharia microalgal na utilização de gás de combustão e extração de biopolímeros para desenvolvimento de nanofibras. Tese de Doutorado em Engenharia e Ciência de Alimentos – FURG, Rio Grande, 2008.

MORAIS, M. G., COSTA, J. A. V. Perfil de ácidos graxos de microalgas cultivadas com dióxido de carbono. *Ciência e Agrotecnologia*. v. 32 (4), p. 1245-1251, 2008a.

MORAIS, M. G., COSTA, J. A. V. Photo bioreactor tubular for removing or fixing of carbon dioxide, greenhouse gases using micro algae and cyan bacteria, involves reducing emissions of main gás responsible for global warming and other gases of greenhouse effect. Patent Number: BR200702736-A, 2008b.

MUSSI, R.; DIMURO, G.P.; BEDREGAL, B.C.; Número Fuzzy Intervalares, Anais XXXII Congresso Nacional de Matemática Aplicada e Computacional, Cuiabá, 2009.

NAKAS, J. P.; SCHAEDELE, M.; PARKINSON, C.M.; COONLEY, C. E.; TANENBAUM, S. W.; Production of solvents from algal biomass. *Appl. Environ. Microbiol.*, v. 46, n. 5, p. 1017-1023, 1983.

OGATA, K; Engenharia de Controle Moderno; Ed. Prentice Hall, 2003.

ORTEGA-ZAMORANO, F., SUBIRATS, J.L., JEREZ, J.M., MOLINA, I., FRANCO, L., Implementation of the C-Mantec neural network constructive algorithm in an arduino UNO microcontroller. *Lect. Notes Comput. Sci.* 7902, 80–87, 2013.

PARAMAR, A.; SINGH, N.; PANDEY, A.; GNANSOUNOU, E.; MADAMWAR, D., Cyanobacteria and Microalgae: A Positive Prospect for Biofuels. *Bioresource Technology*, 102, 22, 10163-10172, 2011

PEDRYCZ, W., Fuzzy Control and Fuzzy Systems. 2nd Ed., John Wiley and Sons, 1993.

PELEG, M.; CORRADINI, M.; NORMAND, M.; The Logistic (Verhulst) Model for Sigmoid Microbial Growth Curves Revisited. *Food Research International* 40, 808-818, 2007.

PENG, X., LIU, S., YAMAKAWA, T., WANG, P., LIU, X., Self-Regulating PID Controllers and its Applications to a Temperature Controlling Process. *Fuzzy Computing*, Elsevier Science Publishers, North-Holland, 1988.

PONTON, J. W., KLEMES, J., Alternatives to Neural Networks for Inferential Measurement. *Computers Chem. Eng.*, v. 17, 10, pp. 991-1000, 1993.

PROCYK, T. J., MAMDANI, E. H., A Linguistic Self-Organising Process Controller. *Automatica*, 15, pp. 15-30, 1979.

PURVIS, M.; Remote Datalogger For Photo-Voltaic Research, Murdoch University, 2013.

RADMANN, E. M. Cultivo de microalgas para produção de biossurfactantes. Tese de Doutorado em Engenharia e Ciência de Alimentos – FURG, Rio Grande, 2011.

RADMANN, E. M. ; CAMERINI, F. V. ; SANTOS, T. D. ; COSTA, J. A. V. Isolation and application of SOX and NOX resistant microalgae in biofixation of CO₂ from thermoelectricity plants. *Energy Conversion and Management*, v. 52, p. 3132-3136, 2011.

RASHID, T. A.; AHMED, H.A.; Building an Adaptive Neural Fuzzy Inference System for Revised General Test/Gross Point Average Problem, *International Journal of Emerging Technologies in Computational and Applied Sciences-IJETCAS*, 12-326, 2012.

RICHMOND, A. - Handbook of Microalgal Culture: Biotechnology and Applied Phycology, 1999.

RICHMOND & TREDICCI; Handbook of Microalgal Culture: Biotechnology and Applied Phycology, 2006.

ROSA, A. P. C. Produção de biomassa e ácidos graxos por diferentes microalgas e condições de cultivo. Tese de Doutorado em Engenharia e Ciência de Alimentos – FURG, Rio Grande, 2012.

SÁNCHEZ, S.; MARTINEZ, M. E.; ESPINOLA, F.; Biomass Production and Biochemical Variability of the Marine Microalgae *Isochysis galbana* in Relation to Culture Medium. *Biochemical Engineering Journal*, 6, 13-18, 2000.

SCHMIDELL, W; AQUARONE, E; BORZANI, W.; *Biotechnologia Industrial*, vol. 2, Edgard Blücher, 2001.

SCHNITMAN, L.; FELIPPE de SOUZA, J.A.M.; Takagi-Sugeno-Kang Fuzzy Structures in Dynamic System Modeling, *ICANN 20th International Conference*, 2010.

SEBORG, D. E.; EDGAR, T.F.; MELLICHAMP, D.; *Process Dynamics and Control*, Wiley, 2 ed., 2003.

SHULER, M.L.; KARGI, F.; Bioprocess Engineering: Basic Concepts, 2a. Ed., Prentice Hall, 2001.

SKOOG, D. A.; HOLLER, F.J.; NIEMAN, T.A.; Principios de Análisis Instrumental; McGraw-Hill Madrid, 5a. Ed.; 2001.

SUGENO, M., Industrial Applications of Fuzzy Control, Elsevier Science Pub. Co., 1985.

STAVREV, V.; Dual-phase flow micro-gravity photobioreactor – Principles and an experimental module design, 41st International Conference on Environmental Systems, Portland, Oregon, 2011.

TAMAGNINI, P.; LEITÃO, E.; OLIVEIRA, P.; FERREIRA, D.; PINTO, F.; HARRIS, D. J. HEIDORN, D.; LINBLAD, P. Cyanobacterial Hydrogenases: Diversity, Regulation and Applications. FEMS Microbiol. Rev., 31, 692–720, 2007.

TREVISAN, M. G.; POPPI, R. J. Process Analytical Chemistry. Química Nova, v. 29, n. 5, p. 1065–1071, 2006.

VEGA, M.; SILVA, J. W.; OLIVEIRA, M. Non Linear Identification of *Spirulina maxima* Growth and Characteristics. Computer Aided Chemical Engineering, 29, 1391-1395, 2011

VELTEN, K.; Mathematical Modeling and Simulation: Introduction for Scientists and Engineers, Wiley-VCH, 2009.

VICENTE, M. J. P.; SARMENTO, M. H. C.; DECT Shield for Arduino, IFT Lisboa, 2012.

VOJINOVIĆ, V.; CABRAL, J. M. S.; FONSECA, L. P. Real-time bioprocess monitoring: Part I: In situ sensors. Sensors and Actuators B: Chemical, v. 114, n. 2, p. 1083–1091, 2006.

VONSHAK, A.; RICHMOND, A. Mass production of the blue-green alga *Spirulina*: An overview. Biomass, v.15, 233-247, 1988.

YAGER, R. R., FILEV, D. P., Essentials of Fuzzy Modeling and Control. 1st Ed., John Wiley & Sons Inc., 388 p., 1994.

YAGISHITA, O., ITOH, O., SUGENO, M., Application of Fuzzy Reasoning to the Water Purification Process. Industrial Applications of Fuzzy Control. M. Sugeno (ed.), Elsevier Science Publishers, North-Holland, 1985.



WALI, W.A.; AL-SHAMMA´A, A.I.; HASSAN, K.H.; CULLEN, J.D.; Online genetic-ANFIS temperature control for advanced microwave biodiesel reactor, Journal of Process Control 22, 1256– 1272, 2012.

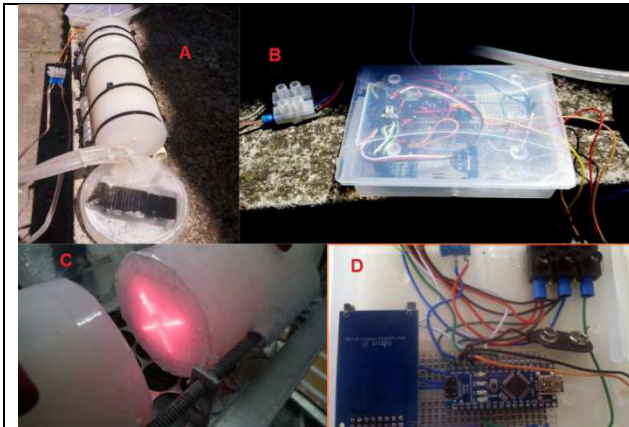
WATANABE, A. List of Algal Strains in Collection at the Institute of Applied Microbiology University of Tokyo. J. Gen. Appl. Microbiol., v. 6, p. 1-4, 1960.

WERNIMONT, G. T.; use of Statistics to Develop and Evaluate Analytical Methods, Edited by W. Spendley, AOAC, Arlington, Virginia, USA, 1985.

- ZADEH, L. A., Fuzzy Sets. *Information and Control*, 8, pp 338-353, 1965.
- ZADEH, L. A., Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Transactions on Systems, Man and Cybernetics*, p 29-44, 1973.
- ZADEH, L.A.; The concept of a Linguistic Variable and its Application to Approximate Reasoning – I, *Information Sciences* 8,199-299, 1975.
- ZADEH, L.A.; The concept of a Linguistic Variable and its Application to Approximate Reasoning – I, *Information Sciences* 8,199-299, 1975.
- ZADEH, L. A., A Theory of Approximate Reasoning. *Machine Intelligence*, v. 9, pp. 149-194. New York, 1979.
- ZADEH, L. A., Fuzzy Logic, Neural Networks, and Soft Computing, *Communication of the ACM*, vol. 37, no 3, pp 77-84, 1994.
- ZANGWILL, A.; *Modern Electrodynamics*, Cambridge University Press, 2012.
- ZARROUK, C. Contribution à l'étude d'une Cyanophycée: influence de divers facteurs physiques et chimiques sur la croissance et la photosynthèse de *Spirulina maxima*. 1966. Tese (Ph.D) - Université Des Paris, Paris, 1966.
- ZITTELLI, G. C., PASTORELLI R., TREDICI M. R. A modular flat panel photobioreactor (MFPP) for indoor mass cultivation of *Nannochloopsis sp.* under artificial illumination. *Journal of Applied Phycology*, v. 12, p 521 – 526, 2000.

APÊNDICE A – Fotografias das Montagens Utilizadas e dos Protótipos dos MARKS

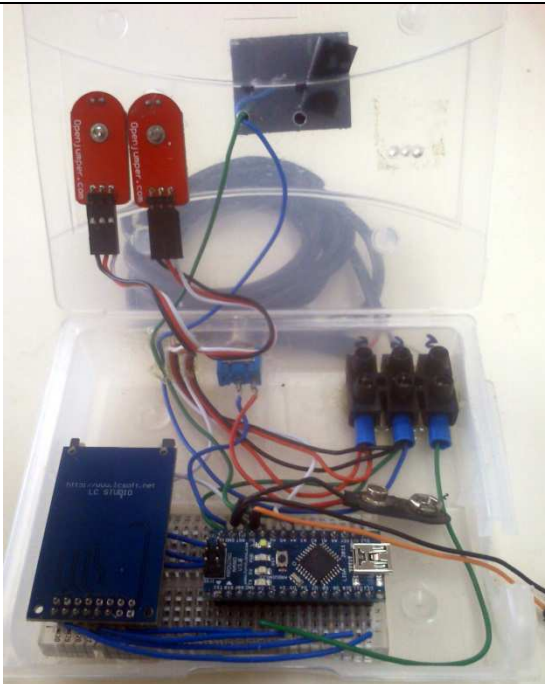
	<p>I - Sistema Óptico 1.0</p>
	<p>II - Instalação do Sistema Óptico 1.0 no Interior do Fotobiorreator <i>Raceway</i> em Estufa <i>Outdoor</i>.</p>



III - (A) Sistema Óptico 1.0 Durante Teste; (B) MARK 1.0; (C) Detalhe do Acionamento do Laser de 660 nm no Sistema Óptico 1.0; (D) Interior do MARK 1.0.

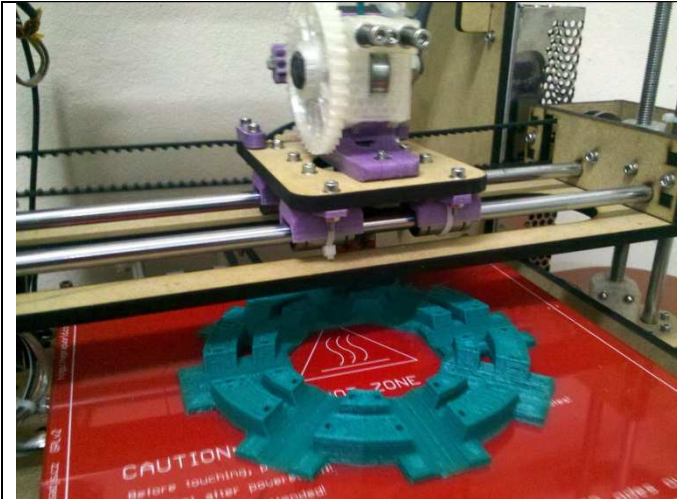


IV - Exterior do MARK 1.0

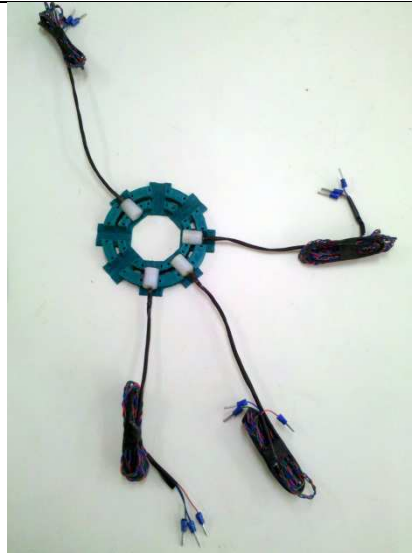


V - Detalhe da Montagem Interna do MARK 1.0

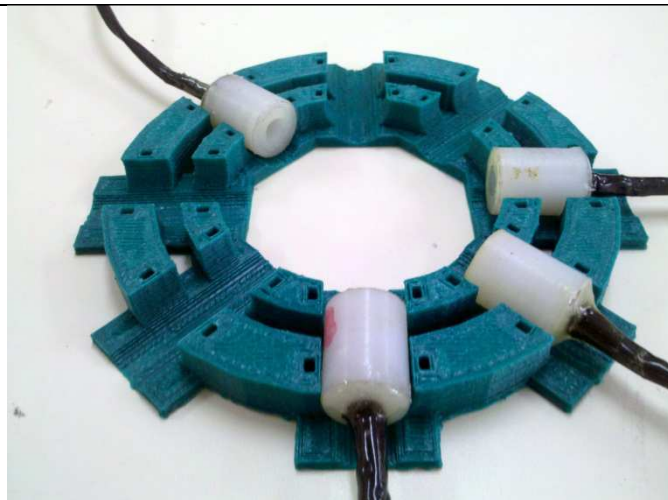
 A photograph showing a custom-built electronic control system for microalgae cultivation. It features a black cooling fan, a clear plastic enclosure housing a microcontroller board, and a white terminal block with several colored wires connected to it. The entire setup is mounted on a wooden surface.	<p>VI - Protótipo do Sistema Autônomo de Monitoramento do Cultivo de Microalgas – MARK 2.5</p>
 A photograph of a white, U-shaped photobioreactor (raceway) filled with a dark green liquid. A central white vertical bar is visible, and various wires and sensors are connected to the liquid. The reactor is placed on a wooden table in a laboratory setting.	<p>VII - Fotobiorreator <i>raceway</i> instrumentado com o Sistema Óptico 1.0 e MARK 2.5, somente com inóculo, sem completar volume com meio de cultivo.</p>
 A photograph of a 3D printer, specifically a Mendel II model. The printer is constructed from wood and metal, with a red filament being extruded from the nozzle. A red printed part is visible on the build platform. The printer is sitting on a white surface.	<p>VIII - Impressora 3D do tipo REPRAP – Mendel II – (Metamáquina – São Paulo)</p>



IX – Impressão do suporte em plástico ABS



X – Suporte com parte dos receptores e com somente um emissor óptico..



XI – Detalhe do suporte com emissor e receptores, parcialmente montado.

APÊNDICE B – Código do Programa do Microcontrolador ARDUINO - MARK 3.0

```

//Código do Datalogger Mark3 em 09_11_2012 por RDPF - versão ZERO

//PA0 Externa - Verde || PA1 RECEPTOR - Verde || PA2 Externa - Vermelha || PA3 RECEPTOR - Verde ||

//PA4 e PA5 Barramento I2C (luxímetro e RTC) - (verificar endereçamento)

//PA6 - Temperatura Interna com LM35

//Digitais

//PD0 RX pH1 - PD1 TX pH1

//PD2 - Barramento OneWire - Temperaturas Internas Meio de Cultivo 1 e 2 (verificar endereçamento)

//PD3 LED de funcionamento

//PD4 - CS - Ocupado pelo Gravador de Cartão SD

//PD5 - Laser Emissor Verde

//PD6 - Laser Emissor Vermelho

//PD7 - RX pH2 - PD8 TX pH2

//PD9 - Saida Relay1

//PD10 - Saida Relay1

//PD11, PD12,PD13 - Gravador de Cartão SD

//const int chipSelect = 4;

// * SD card attached to SPI bus as follows: ** MOSI - pin 11 ** MISO - pin 12 ** CLK - pin 13 ** CS - pin
4

//#include <SdFat.h> // biblioteca para gravar do SD

#include <OneWire.h> // biblioteca para comunicar OneWire - Sensores de Temperatura Dallas

#include <DallasTemperature.h>

//#include <SoftwareSerial.h> // biblioteca para comunicar com os pHmetros

#include <Wire.h> // biblioteca para usar o barramento I2C (usado para o Relógio de Tempo Real e o
Luxímetro )

#define DS1307_I2C_ADDRESS 0x68 // endereçamento I2C do Relógio de Tempo Real

//SdFat sd;

//SdFile myFile;

// Cabo de dados barramento OneWire está plugado na porta 2 do Arduino

#define ONE_WIRE_BUS 2 //check

//diretivas referentes luxímetro

#define DS1307_I2C_ADDRESS 0x68 // Este é o endereçamento I2C

int BH1750address = 0x23;

```

```

byte buff[2];

// diretivas de compatibilidade de versão da IDE do Arduino
#if defined(ARDUINO) && ARDUINO >= 100 // Arduino v1.0 e mais novos
  #define I2C_WRITE Wire.write
  #define I2C_READ Wire.read
#else // Arduino anterior ao v1.0
  #define I2C_WRITE Wire.send
  #define I2C_READ Wire.receive
#endif

// Variaveis Globais

int command = 0; // Esta eh o char de comando, em ASCII, enviado via porta serial

int i;

//variáveis do RTC - mantidas em ling. Inglesa
byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;

byte test;

byte zero;

char *Day[] = { "", "Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat" };
char *Mon[] = { "", "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec" };

//variável vetor de passagem do tempo para gravação
float tempo[7];

// cria uma instancia OneWire para se comunicar com quaisquer dispositivos OneWire
OneWire oneWire(ONE_WIRE_BUS);

// passa a referencia oneWire para os sensores de temperatura Dallas
DallasTemperature sensors(&oneWire);

//variaveis temporarias de recuperacao das temperaturas
float tempD, tempE;

// matrizes para armazenar as temperaturas (mantidas em ling inglesa)
DeviceAddress insideThermometer, outsideThermometer;

char filename[] = "LIGHT000.txt";

int ii=0;

```



```

float h=0,th=0;

int executa=0;

int ativar=1;

float somat=0;

float temp=0;

float tempinterna=0;

unsigned long last=0;

unsigned long last2=0;

int ligado;

float val1=0.0,val2=0.0;

int val3=0;

int valext=0;

int valint=0;    // variável a guardar o valor proveniente do sensor

int leitextg[100];

int leitintg[100];

int leitextr[100];

int leitintr[100];

//int i=0;

int soma1=0;

int soma2=0;

//unsigned long tempo; // retirar

//Definicao dos respectivos pinos digitais de saída

int ledonoff=3;

int lasg=10; //acionamento pelo relay do laser VERDE (G)

int lasr=9; // acionamento pelo relay do laser VERMELHO (R)

int relay1=9; // mantido como na versão original

int relay2=10; // mantido como na versão original

//Definicao dos respectivos pinos analogicos (entrada)

int gext=0; //pino da celula externa do laser verde

int grec=1; //pino da celula interna do laser verde

int rext=2; //pino da celula externa do laser vermelho

int rrec=3; //pino da celula interna do laser vermelho

```

```

//definicao do char para ler os valores de pH e pH2

int peaga1[6],peaga2[6];

int tempPin=6; // pino da temperatura interna do Mark

//SoftwareSerial pH = SoftwareSerial(0, 1); //setup and rename soft uart.
//
//          RX|TX

//SoftwareSerial pH2 = SoftwareSerial(7, 8); //setup and rename soft uart.
//
//          RX|TX

// Convert normal decimal numbers to binary coded decimal
byte decToBcd(byte val)
{
  return ( (val/10*16) + (val% 10) );
}

// Convert binary coded decimal to normal decimal numbers
byte bcdToDec(byte val)
{
  return ( (val/16*10) + (val% 16) );
}

// Gets the date and time from the ds1307 and prints result
void getDateDs1307()
{
  // Reset the register pointer
  Wire.beginTransmission(DS1307_I2C_ADDRESS);
  I2C_WRITE(zero);
  Wire.endTransmission();
  Wire.requestFrom(DS1307_I2C_ADDRESS, 7);
  // A few of these need masks because certain bits are control bits
  second  = bcdToDec(I2C_READ() & 0x7f);
  minute  = bcdToDec(I2C_READ());
  hour    = bcdToDec(I2C_READ() & 0x3f); // Need to change this if 12 hour am/pm
  dayOfWeek = bcdToDec(I2C_READ());
}

```

```

dayOfMonth = bcdToDec(I2C_READ());

month    = bcdToDec(I2C_READ());

year     = bcdToDec(I2C_READ());

// resolver/resolvido usando vetor tempo[7] = aqui questao formatacao do tempo OK!

if (hour < 10)
    Serial.print("0");
Serial.print(hour, DEC);
Serial.print(":");
if (minute < 10)
    Serial.print("0");
Serial.print(minute, DEC);
Serial.print(":");
if (second < 10)
    Serial.print("0");
Serial.print(second, DEC);
Serial.print(" ");
Serial.print(Day[dayOfWeek]);
Serial.print(", ");
Serial.print(dayOfMonth, DEC);
Serial.print(" ");
Serial.print(Mon[month]);
Serial.print(" 20");
if (year < 10)
    Serial.print("0");
Serial.println(year, DEC);
}

// INICIO DO SETUP
void setup() {
// pH.begin(38400); //inicializa comunicacao serial com o pH1
// pH2.begin(38400); //inicializa comunicacao serial com o pH2
// Se executar o Setup cria cabecario - ou seja cada vez que reinicializar
// if (!sd.begin(chipSelect, SPI_HALF_SPEED)) sd.initErrorHalt();

```

```

// if the file opened okay, write to it:

// open the file for write at end like the Native SD library
// if (!myFile.open(filename, O_RDWR | O_CREAT | O_AT_END)) {
//   sd.errorHalt("opening light.txt for write failed");

/*

//resolver cabecario - fazer cabecario correto
myFile.println("-----INICIO-----");
myFile.print("Tempomiliseg");
myFile.print(",");
myFile.print("hora");
myFile.print(",");
myFile.print("min");
myFile.print(",");
myFile.print("seg");
myFile.print(",");
myFile.print("dia");
myFile.print(",");
myFile.print("mes");
myFile.print(",");
myFile.print("ano");
myFile.print(",");
myFile.print("Ativado");
myFile.print(",");
myFile.print("TempMARK");
myFile.print(",");
myFile.print("TempCult1");
myFile.print(",");
myFile.print("TempCult2");
myFile.print(",");
myFile.print("Iluminancia");

```

```

myFile.print(",");
myFile.print("100sensorexternoGREEN");
myFile.print(",");
myFile.print("100sensorinternoGREEN");
myFile.print(",");
myFile.print("100sensorexternoRED");
myFile.print(",");
myFile.print("100sensorinternoRED");
myFile.println("");
myFile.close();
*/

// Final da Rotina de impressão do cabecario do arquivo
Wire.begin();
// ativa a porta serial
Serial.begin(9600);
zero=0x00;
// ativa a biblioteca de sensores
sensors.begin();
// procura pelos sensores no barramento e vincula a um indice
if (!sensors.getAddress(insideThermometer, 0)) Serial.println("Nao foi possivel encontrar enderecamento do
Dispositivo 0");
if (!sensors.getAddress(outsideThermometer, 1)) Serial.println("Nao foi possivel encontrar enderecamento do
Dispositivo 0");
for (int i = 0; i < 100; i++){
  leitextg[i] = 0;}
for (int i = 0; i < 100; i++){
  leitintg[i] = 0;}
for (int i = 0; i < 100; i++){
  leitextr[i] = 0;}
for (int i = 0; i < 100; i++){
  leitintr[i] = 0;}

//Definicao dos respectivos pinos digitais de saída
pinMode(ledonoff, OUTPUT);

```

```

pinMode(lasg, OUTPUT);
pinMode(lasr, OUTPUT);
pinMode(relay1, OUTPUT);
pinMode(relay2, OUTPUT);
}
void loop() {
    float teste;
//=====
//if (millis() - last > 10000) // faz leituras de 5 em 5 minutos RESOLVER
//{
    last=millis();
    //novo codigo dos sensores Dallas, do RTC e do Luximetro
// obtem as medidas de temperatura dos sensores
sensors.requestTemperatures();
// atribue as temperaturas para as variaveis a serem manipuladas pelo restante do programa do Mark
tempD = sensors.getTempC(insideThermometer);
tempE = sensors.getTempC(outsideThermometer);

//Somente para fins de verificacao em saida serial
Serial.println();
Serial.println();
Serial.println(tempD);
Serial.println(tempE);
    delay(1000);
    // RTC e Luximetro
//    getDateDs1307();
//    delay(100);
//
//=====
//luximetro
//
//=====

```

```

    int i;

    uint16_t val=0;

    BH1750_Init(BH1750address);

    delay(200);

    if(2==BH1750_Read(BH1750address))
    {
        val=((buff[0]<<8)|buff[1])/1.2;

        Serial.print(val,DEC);

        Serial.print("[lx]");

        //Serial.println(val);

        teste=val;

        Serial.println(teste);

    }

    delay(150);

    //

    //if (tempo[4] !=0.0) {

    // Serial.println("TEMPO RECUPERADO");

    // Serial.println(tempo[0]);

    // Serial.println(tempo[1]);

    // Serial.println(tempo[2]);

    // Serial.println(tempo[3]);

    // Serial.println(tempo[4]);

    // Serial.println(tempo[5]);

    // Serial.println(tempo[6]);

    // Serial.println(tempo[7]);

    }

//Fim do novo codigo dos Sensores Dallas, do RTC e do Luximetro

/*

```

```

//
=====

// =====

// Fim do RTC e do Luximetro

// =====

// Inicio dos pHmetros
pH.print("r"); //comando de leitura de 1 valor
pH.print((char)13); // return em ASCII
delay(1000); //delay para estaabilizar para medida
char inByte[2]; //variavel a ser lida
inByte[1]='0';inByte[0]='0';
//inByte = pH.read();
while(pH.available() > 0)
{
  Serial.print("pH1_available: ");
  Serial.println(pH.available());

  inByte[0] = pH.read();
  peaga1[0]=atoi(inByte);
  Serial.print(inByte[0]);

  inByte[0] = pH.read();
  peaga1[1]=atoi(inByte);
  Serial.print(inByte[0]);

  inByte[0] = pH.read();
  peaga1[2]=atoi(inByte);
  Serial.print(inByte[0]);

  inByte[0] = pH.read();
  peaga1[3]=atoi(inByte);
  Serial.print(inByte[0]);
}

```



```

inByte[0] = pH.read();
peaga1[4]=atoi(inByte);

Serial.println(inByte[0]);

} // serah que precisa de 5 leituras ?
pH.print("e");    //comando de stop (End)
pH.print((char)13); // return em ASCII

/*
// Segundo pHmetro
pH2.print("r");
pH2.print((char)13);
delay(1000);
//char inByte = '0';
//inByte = pH.read();
while(pH2.available() > 0)
{
    Serial.print("pH_available: ");
    Serial.println(pH2.available());

    inByte = pH2.read();
    Serial.print(inByte);
    inByte = pH2.read();
    Serial.print(inByte);
    inByte = pH2.read();
    Serial.print(inByte);
    inByte = pH2.read();
    Serial.print(inByte);
    inByte = pH2.read();
    Serial.print(inByte);
    inByte = pH2.read();
    Serial.println(inByte);
} // serah que precisa de 5 leituras ?
pH2.print("e");    //comando de stop (End)

```

```

pH2.print((char)13); // return em ASCII

//Final da seccao dos pHmetros

*/

//  somat=0;

// LACOS DE MEDICOES DE TEMPERATURAS NOS TANQUES COM SENSORES DB18S20
resolver/resolvido com OneWire e variaveis tempD e tempE ok!

/*

//Variaveis com as temperaturas dos Cultivos: tempD e tempE

// Acionamento de Relays baseado nas temperaturas dos cultivos resolver - pendente - depende do tipo de
acionamento desejado

if (temp>25){

    digitalWrite(6,HIGH);

    ligado=1;

}

if (temp<25){

    digitalWrite(6,LOW);

    ligado=0;

}

*/

// Final da secção dos relays

//  soma1=0;soma2=0;

//PA0 Externa - Verde || PA1 RECEPTOR - Verde || PA2 Externa - Vermelha || PA3 RECEPTOR - Verde ||

// gext=0, grec=1,rect=2,rrec=3;

// Leituras dos sensores externos de luz para biomassa

for (int i = 0; i < 100; i++){

    valext = analogRead(gext); // leitura da celula de silicio1

    leitextg[i] = valext; // armazenar leituras do sensor luz meio biomassa

}

for (int i = 0; i < 100; i++){

    valext = analogRead(rect); // leitura da celula de silicio1

    leitextr[i] = valext; // armazenar leituras do sensor luz meio biomassa

```

```

}

// ativação do laser verde
digitalWrite(lasg, HIGH); // liga o laser verde
delay(1000);           //espera estabilizar
for (int i = 0; i < 100; i++){
  valint = analogRead(grec); // leitura da fotocelula interna do meio sensor de biomassa laser verde
  Serial.println(valint);
  leitintg[i]=valint;
}
digitalWrite(lasg, LOW); // desliga o laser verde

// Laser Vermelho
// ativação do laser vermelho
// digitalWrite(lasr, HIGH); // liga o laser vermelho
// delay(1000);           //espera estabilizar
// for (int i = 0; i < 100; i++){
//   valint = analogRead(rrec); // leitura da fotocelula interna do meio sensor de biomassa laser vermelho
//   leitintr[i]=valint;
// }
// digitalWrite(lasr, LOW); // desliga o laser vermelho
somat=0;
// USA LM35 para medir a temperatura interna do Mark
for (int i=1; i <= 10001; i++){
  float temperature = getVoltage(tempPin); //getting the voltage reading from the temperature sensor
  temperature = (temperature) * 100;      //converting from 10 mv per degree wit 500 mV offset
  somat+=temperature;
}
tempinterna=somat/10000;
/*
// secção de gravação em SD
if (!sd.begin(chipSelect, SPI_HALF_SPEED)) sd.initErrorHalt();
// se o arquivo foi aberto, escreva-o:

```

```

// abra o arquivo para escrita no fim como a library SD nativa
if (!myFile.open(filename, O_RDWR | O_CREAT | O_AT_END)) {
  sd.errorHalt("opening file for write failed");
}

//Secção de efetiva escrita no SD - Falta verificar sequencia de gravacao - resolver
myFile.print(last); //contador usando variável interna millis
myFile.print(",");

myFile.print(tempo[2]); // horas do RTC resolver
myFile.print(",");
myFile.print(tempo[1]); // minuto do RTC resolver
myFile.print(",");
myFile.print(tempo[0]); // segundo do RTC resolver
myFile.print(",");
myFile.print(tempo[4]); // dia do RTC resolver
myFile.print(",");
myFile.print(tempo[5]); // mes do RTC resolver
myFile.print(",");
myFile.print(tempo[6]); // ano do RTC resolver
myFile.print(",");
myFile.print(ligado); //variavel logica do acionamento dos reles
myFile.print(",");
myFile.print(tempinterna); //temperatura interna do Mark
myFile.print(",");
myFile.print(tempD); //temperatura em 1 dos cultivos
myFile.print(",");
myFile.print(tempE); //temperatura no outro cultivo
myFile.print(",");
myFile.print(val); //Luximetro Digital em Lux (>106000 = NaN)
myFile.print(",");
for (int i = 0; i < 100; i++){

```

```

    myFile.print(leitextg[i]); //gravacao das medidas externas do laser verde
    myFile.print(",");
}
for (int i = 0; i < 100; i++){
    myFile.print(leitintg[i]); //gravacao das medidas internas do laser verde
    myFile.print(",");
}

for (int i = 0; i < 100; i++){
    myFile.print(leitextr[i]); //gravacao das medidas externas do laser vermelho
    myFile.print(",");
}
for (int i = 0; i < 100; i++){
    myFile.print(leitintr[i]); //gravacao das medidas internas do laser vermelho
    myFile.print(",");
}
myFile.println("");

myFile.close();
} // final do laço de 5 minutos

*/

// Ligar LED para dizer que o funcionamento está OK !

if (ativar==1){
    digitalWrite(ledonoff, HIGH);
    delay(1000);
    digitalWrite(ledonoff, LOW);
    delay(1000);
}

if (ativar==0){
    digitalWrite(ledonoff, HIGH);

```

```

}

if (millis()-last>295000){
  ativar=0;}

if (millis()-last<295000){
  ativar=1;}

Serial.println("");
Serial.print("MEMORIA SRAM: ");
Serial.println(freeRAM(), DEC);
}
//}

//FINAL DO PROGRAMA PRINCIPAL
//Rotina para verificacao da memoria livre
int freeRAM() {
extern int __heap_start, *__brkval;
int v;
return (int) &v - (__brkval == 0 ? (int) &__heap_start : (int) __brkval);
}

/*
* getVoltage() - returns the voltage on the analog input defined by
* pin
*/
float getVoltage(int pin){
return (analogRead(pin) * .004882814); //converting from a 0 to 1023 digital range
// to 0 to 5 volts (each 1 reading equals ~ 5 millivolts
}

//=====
int BH1750_Read(int address) //

```

```

{
int i=0;
Wire.beginTransmission(address);
Wire.requestFrom(address, 2);
while(Wire.available() //
{
buff[i] = Wire.read(); // receive one byte
i++;
}
Wire.endTransmission();
return i;
}
void BH1750_Init(int address)
{
Wire.beginTransmission(address);
Wire.write(0x10); // 1x resolution 120ms
Wire.endTransmission();
}
// codigos novos dos Dalas, RTC e Luximetro
//diretivas referentes OneWire

// -----fim do codigo principal

// mais subrotinas
//int BH1750_Read(int address) //
//{{
//int i=0;
//Wire.beginTransmission(address);
//Wire.requestFrom(address, 2);
//while(Wire.available() //
//{{
//buff[i] = Wire.read(); // receive one byte
//i++;

```

```
//}  
  
//Wire.endTransmission();  
  
//return i;  
//}  
  
  
//void BH1750_Init(int address)  
  
//{  
//Wire.beginTransmission(address);  
//Wire.write(0x10);//11x resolution 120ms  
//Wire.endTransmission();  
//}  
  
  
// Subroutines  
  
// Convert normal decimal numbers to binary coded decimal  
//byte decToBcd(byte val)  
  
//{  
// return ( (val/10*16) + (val%10) );  
//}  
  
  
// Convert binary coded decimal to normal decimal numbers  
//byte bcdToDec(byte val)  
  
//{  
// return ( (val/16*10) + (val%16) );  
//}  
  
  
// 1) Sets the date and time on the ds1307  
// 2) Starts the clock  
// 3) Sets hour mode to 24 hour clock  
// Assumes you're passing in valid numbers, Probably need to put in checks for valid numbers.  
  
void setDateDs1307()  
  
{
```



```

    second = (byte) ((Serial.read() - 48) * 10 + (Serial.read() - 48)); // Use of (byte) type casting and ascii math to
    achieve result.

```

```

    minute = (byte) ((Serial.read() - 48) * 10 + (Serial.read() - 48));

```

```

    hour = (byte) ((Serial.read() - 48) * 10 + (Serial.read() - 48));

```

```

    dayOfWeek = (byte) (Serial.read() - 48);

```

```

    dayOfMonth = (byte) ((Serial.read() - 48) * 10 + (Serial.read() - 48));

```

```

    month = (byte) ((Serial.read() - 48) * 10 + (Serial.read() - 48));

```

```

    year= (byte) ((Serial.read() - 48) * 10 + (Serial.read() - 48));

```

```

    Wire.beginTransmission(DS1307_I2C_ADDRESS);

```

```

    I2C_WRITE(zero);

```

```

    I2C_WRITE(decToBcd(second) & 0x7f); // 0 to bit 7 starts the clock

```

```

    I2C_WRITE(decToBcd(minute));

```

```

    I2C_WRITE(decToBcd(hour)); // If you want 12 hour am/pm you need to set

```

```

        // bit 6 (also need to change readDateDs1307)

```

```

    I2C_WRITE(decToBcd(dayOfWeek));

```

```

    I2C_WRITE(decToBcd(dayOfMonth));

```

```

    I2C_WRITE(decToBcd(month));

```

```

    I2C_WRITE(decToBcd(year));

```

```

    Wire.endTransmission();

```

```

}

```

```

// Gets the date and time from the ds1307 and prints result

```

```

//void getDateDs1307()

```

```

//{{

```

```

// // Reset the register pointer

```

```

// Wire.beginTransmission(DS1307_I2C_ADDRESS);

```

```

// I2C_WRITE(zero);

```

```

// Wire.endTransmission();

```

```

//

```

```

// Wire.requestFrom(DS1307_I2C_ADDRESS, 7);

```

```

//

```

```

// // A few of these need masks because certain bits are control bits

```

```

// second = bcdToDec(I2C_READ() & 0x7f);

```

```

// minute = bcdToDec(I2C_READ());
// hour = bcdToDec(I2C_READ() & 0x3f); // Need to change this if 12 hour am/pm
// dayOfWeek = bcdToDec(I2C_READ());
// dayOfMonth = bcdToDec(I2C_READ());
// month = bcdToDec(I2C_READ());
// year = bcdToDec(I2C_READ());
//
//
tempo[0]=second;tempo[1]=minute;tempo[2]=hour;tempo[3]=hour;tempo[4]=dayOfWeek;tempo[5]=dayOfMon
th;tempo[6]=month;tempo[7]=year;
//
// if (hour < 10)
// Serial.print("0");
// Serial.print(hour, DEC);
// Serial.print(":");
// if (minute < 10)
// Serial.print("0");
// Serial.print(minute, DEC);
// Serial.print(":");
// if (second < 10)
// Serial.print("0");
// Serial.print(second, DEC);
// Serial.print(" ");
// Serial.print(Day[dayOfWeek]);
// Serial.print(", ");
// Serial.print(dayOfMonth, DEC);
// Serial.print(" ");
// Serial.print(Mon[month]);
// Serial.print(" 20");
// if (year < 10)
// Serial.print("0");
// Serial.println(year, DEC);
//

```

```
//}
```

```
// Código do MARK 2.0 – POR RDPF
```

```
//celula externa
```

```
//jumper laranja = ground
```

```
//jumper vermelho = analogica 6
```

```
//celula interna
```

```
//jumper laranja = ground
```

```
//jumper amarelo = analogica 5
```

```
//ativacao do Laser
```

```
//jumper azul = ground
```

```
//jumper vermelho = digital 5 como saida
```

```
/* YourDuino.com Example Software Sketch
```

```
  DHT11 Humidity and Temperature Sensor test
```

```
  Credits: Rob Tillaart
```

```
  http://arduino-direct.com/sunshop/index.php?l=product\_detail&p=162
```

```
  terry@yourduino.com */
```

```
/*-----( Import needed libraries )-----*/
```

```
#include <dht11.h>
```

```
/*-----( Declare objects )-----*/
```

```
dht11 DHT11;
```

```
/*-----( Declare Constants, Pin Numbers )-----*/
```

```
#define DHT11PIN 2
```

```

int temperaturePin = 0; //the analog pin the TMP36's Vout (sense) pin is connected to
    //the resolution is 10 mV / degree centigrade
    //(500 mV offset) to make negative temperatures an option

float soma=0;

int luz;

float getVoltage(int pin){
    return (analogRead(pin) * .004882814); //converting from a 0 to 1023 digital range
        // to 0 to 5 volts (each 1 reading equals ~ 5 millivolts
    }

const int chipSelect = 10;

#define leituras1 300 //numero de leituras da celula externa
#define leituras2 300 //numero de leituras da celula interna

#include <SdFat.h>

SdFat sd;

SdFile myFile;

int analogext = 6; //pino da leitura da fotocelula externa
int valext = 0; //variavel da leitura da fotocelula externa
int laserpin = 5; // laser conectado a digital 5
int vezes,i; //variaveis indices
int analogint=5; //pino da leitura da fotocelula interna
int valint=0; //variavel da leitura da fotocelula interna
int leitext[leituras1]; //vetor de leituras externas
int leitint[leituras2]; //vetor de leituras internas
volatile unsigned long cnt = 0;
unsigned long oldcnt = 0;
unsigned long t = 0;
unsigned long last;

//int potPin = 4; // selecione o pino de entrada da fotocelula
//int val=0; // variável a guardar o valor proveniente do sensor

```

```
void irq1()
{
  cnt++;
}

void setup()
{

  Serial.begin(9600);
  //Serial.println("DHT11 TEST PROGRAM ");
  //Serial.print("LIBRARY VERSION: ");
  //Serial.println(DHT11LIB_VERSION);
  //Serial.println();
}/*--(end setup)---*/

void loop() /*----( LOOP: RUNS CONSTANTLY )----*/
{
  // Serial.println("\n");

  int chk = DHT11.read(DHT11PIN);

  // Serial.print("Read sensor: ");
  // switch (chk)
  // {
  //   case 0: Serial.println("OK"); break;
  //   case -1: Serial.println("Checksum error"); break;
  //   case -2: Serial.println("Time out error"); break;
  //   default: Serial.println("Unknown error"); break;
  // }

  // UMIDADE

  // Serial.print("Humidity (%): ");
```

```

// Serial.println((float)DHT11.humidity, 2);

// TEMPERATURA C

// Serial.print("Temperature (oC): ");

// Serial.println((float)DHT11.temperature, 2);

Serial.print("Temperature (oF): ");

Serial.println(Fahrenheit(DHT11.temperature), 2);

Serial.print("Temperature (K): ");

Serial.println(Kelvin(DHT11.temperature), 2);

Serial.print("Dew Point (oC): ");

Serial.println(dewPoint(DHT11.temperature, DHT11.humidity));

Serial.print("Dew PointFast (oC): ");

Serial.println(dewPointFast(DHT11.temperature, DHT11.humidity));

delay(2000);

while (i<5000){

float temperature = getVoltage(temperaturePin); //getting the voltage reading from the temperature sensor

temperature = (temperature - .5) * 100; //converting from 10 mv per degree wit 500 mV offset

soma=soma+temperature;

i=i++;

}

i=0;

//to degrees ((volatge - 500mV) times 100)

Serial.println(soma/5000); //printing the result

delay(100); //waiting a second

soma=0;

//luz

luz = analogRead(3);

Serial.println(luz);

```

```

// DATALOGGER
pinMode(2, INPUT);
digitalWrite(2, HIGH);
attachInterrupt(0, irq1, RISING);
Serial.begin(9600);    // setup serial
pinMode(laserpin, OUTPUT); // define o pino laser como saida
for (int i = 0; i < leituras1; i++){
    leitext[i] = 0;}
for (int i = 0; i < leituras2; i++){
    leitint[i] = 0;}
if (millis() - last > 60000)
{
    last = millis();

for (int i = 0; i < leituras1; i++){
    valext = analogRead(analogext); // leitura da fotocelula externa
    leitext[i] = valext; // armazenar leituras do sensor luz externa
}
digitalWrite(laserpin, HIGH); // liga o laser
delay(3000);
for (int i = 0; i < leituras2; i++){
    delay(10000/300);
    valint = analogRead(analogint); // leitura da fotocelula interna
    leitint[i]=valint;
}
digitalWrite(laserpin, LOW); // desliga o laser
//Serial.println(valext);    //mostra valor da fotocelula externa
//Serial.println(valint);    //mostra valor da fotocelula interna
// for (int i = 0; i < leituras1; i++){
//   Serial.println(leitext[i]);
// }

```

```

// for (int i = 0; i < leituras2; i++){
//   Serial.println(leitint[i]);
// }

//Gravacao em SD
if (!sd.begin(chipSelect, SPI_HALF_SPEED)) sd.initErrorHalt();

// if the file opened okay, write to it:
//   val = analogRead(potPin); // ler o valor da fotocelula
//   Serial.print(val);
//   Serial.print("\t");

   t = cnt;

   unsigned long hz = t - oldcnt;
//   Serial.print("FREQ: ");
//   Serial.print(hz);
//   Serial.print("\t = ");
//   Serial.print((hz+50)/100); // +50 == rounding last digit
//   Serial.println(" mW/m2");

   oldcnt = t;

// open the file for write at end like the Native SD library
if (!myFile.open("light.txt", O_RDWR | O_CREAT | O_AT_END)) {
   sd.errorHalt("opening light.txt for write failed");
}

myFile.print(last);
myFile.print(",");
myFile.print(hz);
myFile.print(",");
// myFile.print(val);
// myFile.print(",");

for (int i = 0; i < leituras1; i++){
   myFile.print(leitext[i]);
   myFile.print(",");
}

```



```
for (int i = 0; i < leituras2; i++){  
    myFile.print(leitint[i]);  
    myFile.print(",");  
}  
myFile.println("");  
myFile.close();  
}
```

```
//FIM DO CÓDIGO //
```

APÊNDICE C – PROGRAMA DO ARDUINO DUE PARA MEDIÇÃO DA INTENSIDADE DA RADIAÇÃO MONOCROMÁTICA ESPALHADA (SCATTERING) PELA SUSPENSÃO DE *SPIRULINA*

```
// por RDPF – janeiro 2013.

int ativa_relay,fita,laser;

int contador1=0;

int total=1;

int vez=1;

unsigned long t1 = 0;

//unsigned long t2 = 0;

//unsigned long t3 = 0;

//unsigned long t4 = 0;

unsigned long t5 = 0;

volatile unsigned long cnt1 = 0;

//volatile unsigned long cnt2 = 0;

//volatile unsigned long cnt4 = 0;

//volatile unsigned long cnt5 = 0;

unsigned long oldcnt1 = 0;

//unsigned long oldcnt2 = 0;

//unsigned long oldcnt3 = 0;

//unsigned long oldcnt4 = 0;

//unsigned long oldcnt5 = 0;

unsigned long hz1 = 0;

//unsigned long hz2 = 0;

//unsigned long hz3 = 0;

//unsigned long hz4 = 0;

//unsigned long hz5 = 0;

unsigned long t = 0;

unsigned long last;

//45 Graus para os Dois Emissores

void irq1()
```

```
{
  cnt1++;
}
/*
//135 Graus para os Dois Emissores
void irq2()
{
  cnt2++;
}
//180 Graus do Vermelho e o 90 do Verde - Código 180R na Montagem
void irq4()
{
  cnt4++;
}
//eXTERNA
void irq5()
{
  cnt5++;
}

*/
////////////////////////////////////
//
// SETUP
//
void setup()
{
  Serial.begin(115200);
  Serial.println("START");

  pinMode(37,INPUT);
// pinMode(33,INPUT);
// pinMode(35,INPUT);
```



```

//NO ESCURO

delay(10000);

Serial.println("180G Laser R");

Serial.print("=====INICIO ESCURO=====");

Serial.println();

delay(5000);

for (int j = 1; j < 10001; j++){

  //45 Graus para os Dois Emissores

  t1=cnt1;

  //45 Graus para os Dois Emissores

  // t2=cnt2;

  //180 Graus do Vermelho Verde e o 90 do Verde - Código 180R na Montagem

  // t4=cnt4;

  // t5=cnt5; //Externa

  unsigned long hz1 = t1 - oldcnt1;

  // unsigned long hz2 = t2 - oldcnt2;

  // unsigned long hz4 = t4 - oldcnt4;

  // unsigned long hz5 = t5 - oldcnt5;

  // Display na Serial as leituras dos sensores

  Serial.print(vez);Serial.print(',');

  Serial.println(hz1);//Serial.print(',');

  // Serial.print(hz2);Serial.print(',');

  // Serial.print(hz4);Serial.print(',');

  // Serial.println(hz5);

  vez=vez+1;

  oldcnt1=t1; //oldcnt2=t2;oldcnt4=t4;oldcnt5=t5;

}

Serial.println("=====FINAL==ESCURO=====");

Serial.println();

// Aqui acabava o teste inicial

fita=40;

```

```

while (fita<45){
  vez=1;
  Serial.println("=====INICIO FITA =====");
  Serial.println(fita-39);
  Serial.println("=====INICIO FITA =====");
  digitalWrite(fita,HIGH);
  cnt1=0;//cnt2=0;cnt4=0;cnt5=0;
  oldcnt1=0;//oldcnt2=0;oldcnt4=0;oldcnt5=0;
  delay(5000);
  for (int j = 1; j < 10001; j++){
    t1=cnt1;
//    t2=cnt2;
//    t4=cnt4;
//    t5=cnt5;
    hz1 = t1 - oldcnt1;
//    hz2 = t2 - oldcnt2;
//    hz4 = t4 - oldcnt4;
//    hz5 = t5 - oldcnt5;
    // Display na Serial as leituras dos sensores
    Serial.print(vez);Serial.print(',');
    Serial.println(hz1);//Serial.print(',');
//    Serial.print(hz2);Serial.print(',');
//    Serial.print(hz4);Serial.print(',');
//    Serial.println(hz5);
    vez=vez+1;
    oldcnt1=t1; //oldcnt2=t2;oldcnt4=t4;oldcnt5=t5;
  }
  fita=fita+1;
}
fita=40;
while (fita<45){
  digitalWrite(fita,LOW);
  fita=fita+1;
}

```

```

}

Serial.println("=====FINAL==FITAS LIGADAS=====");

// VERDE Vermelho

laser=50;

digitalWrite(laser,HIGH);

delay(10000);

vez=1;

//NO ESCURO

delay(5000);

cnt1=0;//cnt2=0;cnt4=0;cnt5=0;

oldcnt1=0;//oldcnt2=0;oldcnt4=0;oldcnt5=0;

Serial.println("=====INICIO LASER VERMELHO NO ESCURO=====");

for (int j = 1; j < 10001; j++){

    t1=cnt1;

//    t2=cnt2;

//    t4=cnt4;

//    t5=cnt5;

    hz1 = t1 - oldcnt1;

//    hz2 = t2 - oldcnt2;

//    hz4 = t4 - oldcnt4;

//    hz5 = t5 - oldcnt5;

// Display na Serial as leituras dos sensores

    Serial.print(vez);Serial.print(',');

    Serial.println(hz1);//Serial.print(',');

//    Serial.print(hz2);Serial.print(',');

//    Serial.print(hz4);Serial.print(',');

//    Serial.println(hz5);

    vez=vez+1;

```

```
    oldcnt1=t1; //oldcnt2=t2;oldcnt4=t4;oldcnt5=t5;
}

fita=40;
while (fita<45){
    vez=1;
    cnt1=0;//cnt2=0;cnt4=0;cnt5=0;
    oldcnt1=0;//oldcnt2=0;oldcnt4=0;oldcnt5=0;
    digitalWrite(fita,HIGH);
    delay(5000);
    for (int j = 1; j < 10001; j++){
        t1=cnt1;
//    t2=cnt2;
//    t4=cnt4;
//    t5=cnt5;
        hz1 = t1 - oldcnt1;
//    hz2 = t2 - oldcnt2;
//    hz4 = t4 - oldcnt4;
//    hz5 = t5 - oldcnt5;
        // Display na Serial as leituras dos sensores
        Serial.print(vez);Serial.print(',');
        Serial.println(hz1);//Serial.print(',');
//    Serial.print(hz2);Serial.print(',');
//    Serial.print(hz4);Serial.print(',');
//    Serial.println(hz5);
        vez=vez+1;
        oldcnt1=t1; //oldcnt2=t2;oldcnt4=t4;oldcnt5=t5;
    }
    fita=fita+1;
}
```



```
fitas=40;
while (fitas<45){
digitalWrite(fitas,LOW);
fitas=fitas+1;
}
digitalWrite(laser,LOW);
Serial.println("FINAL DE TODAS AS MEDIDAS - EXPERIMENTO");
Serial.println(total);
Serial.println("FINAL DE TODAS AS MEDIDAS - EXPERIMENTO");
total=total+1;}
```

APÊNDICE D – PROGRAMA EM SCILAB 5.5 da REDE ANFIS TSK + PSO

```

//// ANFIS - Sistema de Inferência Neurofuzzy Adaptativo para
// soft Sensing de Concentração a partir de dados de
// sensor óptico combinado de biomassa microalgal.
// Treinamento baseado na minimização do erro quadrático
// pelo algoritmo da otimização por enxame de partículas (PSO)
// de Eberhart e Kennedy (1995)
// usando Z(conc em g/l), X e Y, conforme a combinação:
// X(intensidade sinal 135graus) e Y(intensidade sinal 90graus)
// X(intensidade sinal 135graus) e Y(intensidade sinal 90graus)
// X(intensidade sinal 135graus) e Y(intensidade sinal 90graus)
// by RDPF em 19/02/2014

// inter2, interacoes - variáveis de iteracoes - realocar enxame
// tpop - número de partículas do enxame
// erro_q_graf - erro quadrático para graficar ao longo das iteracoes
// erres -coeficiente de determinacao de Pearson ao longo das iteracoes
// fuzzyTS - função que realiza a inferência fuzzy
// tn e tny - x e y normalizados para serem fuzzificados
// xxi e yyi - partições do conjunto fuzzy que descrevem as
// funcoes de pertinências lineares
// perti(i,...) - grau de pertinência de X ao conjunto fuzzy
// perty(i,...) - grau de pertinência de Y ao conjunto fuzzy
// a(jj), b(jj)-> coeficientes da função linear de consequencia da regra
// se-então na arquitetura TSK
// soma - variável que retorna na função Fuzzy TSK

clc;clear
inter2=1;iteracoes=1;tpop=10;erro_q_graf=zeros(tpop);erres=zeros(200);seq=zeros(200);ab=zeros(200);cdd=zer
os(200);xxu=zeros(200);yyu=zeros(200)
vezess=1;divisor=1;
// Funcao Zfuzzy: efetua o processamento da rede ANFIS
function zfuzzy=fuzzyTS(regras, pontox, pontoy, a, b, c, d, xxi, yyi, zzi, fex, fey)
pert=zeros(1,2*regras);perty=zeros(1,2*regras)
dados=1;tn=pontox;tny=pontoy;xxi=fex;yyi=fey;

xxi(1)=0.25;xxi(2)=0.5;xxi(3)=0.75;
yyi(1)=0.25;yyi(2)=0.5;yyi(3)=0.75;

//fuzzificação da entrada X
i=1
if tn>-0.01 & tn<=xxi(1) then
    pert(i,1)=tn/xxi(1);pert(i,2)=1-pert(i,1);
    pert(i,3)=0;pert(i,4)=0;pert(i,5)=0;pert(i,6)=0;pert(i,7)=0;pert(i,8)=0
end
if tn>xxi(1) & tn<=xxi(2) then
    pert(i,3)=(tn-xxi(1))/(xxi(2)-xxi(1)); pert(i,4)=1-pert(i,3)
    pert(i,1)=0;pert(i,2)=0;pert(i,5)=0;pert(i,6)=0;pert(i,7)=0;pert(i,8)=0
end
if tn>xxi(2) & tn<=xxi(3) then
    pert(i,5)=(tn-xxi(2))/(xxi(3)-xxi(2)); pert(i,6)=1-pert(i,5)
    pert(i,1)=0;pert(i,2)=0;pert(i,3)=0;pert(i,4)=0;pert(i,7)=0;pert(i,8)=0
end
if tn>xxi(3) & tn<=1.01 then
    pert(i,7)=(tn-xxi(3))/(1-xxi(3)); pert(i,8)=1-pert(i,7)
    pert(i,1)=0;pert(i,2)=0;pert(i,3)=0;pert(i,4)=0;pert(i,5)=0;pert(i,6)=0
end
//fuzzificacao da entrada y
if tny>-0.01 & tny<=yyi(1) then

```

```

    perty(i,1)=tny/yyi(1); perty(i,2)=1-perty(i,1)
    perty(i,3)=0;perty(i,4)=0;perty(i,5)=0;perty(i,6)=0;perty(i,7)=0;perty(i,8)=0
end
if tny>yyi(1) & tny<=yyi(2) then
    perty(i,3)=(tny-yyi(1))/(yyi(2)-yyi(1));perty(i,4)=1-perty(i,3)
    perty(i,1)=0;perty(i,2)=0;perty(i,5)=0;perty(i,6)=0;perty(i,7)=0;perty(i,8)=0
end
if tny>yyi(2) & tny<=yyi(3) then
    perty(i,5)=(tny-yyi(2))/(yyi(3)-yyi(2)); perty(i,6)=1-perty(i,5)
    perty(i,1)=0;perty(i,2)=0;perty(i,3)=0;perty(i,4)=0;perty(i,7)=0;perty(i,8)=0
end
if tny>yyi(3) & tny<=1.01 then
    perty(i,7)=(tny-yyi(3))/(1-yyi(3));perty(i,8)=1-perty(i,7)
    perty(i,1)=0;perty(i,2)=0;perty(i,3)=0;perty(i,4)=0;perty(i,5)=0;perty(i,6)=0
end
// "motor" de inferencia fuzzy
soma=0;i=1;
for ww=1:8
    for jj=1:8
        coef=1;
//soma=soma+(((coef*(pert(i,jj)*(tn+a(jj)))/(tn+b(jj)))+ coef*(perty(i,ww)*(tny+c(ww))/(tny+d(ww))))))
//
//soma=soma+abs(((coef*(pert(i,jj)*(a(jj))*(tn+b(jj)))+ coef*(perty(i,ww)*(c(ww))*(tny+d(ww))))))
//soma=soma+abs(((a(jj)*pert(i,jj)*(tn))+ c(ww)*perty(i,ww)*(tny)))
//soma=soma+(coef*pert(i,jj)*(a(jj)*(tn^2+b(jj)*tn)))+(coef*perty(i,ww)*(c(jj)*(tny^2+d(jj)*tny)))

//aka=pert(i,jj)+perty(i,ww);aka=1
//disp("aka=")
//disp(aka)
aka=1;
if aka>0 then
if pert(i,jj)<perty(i,ww) then
    soma=soma+abs(((coef*(pert(i,jj)*(a(jj))*(tn+b(jj)))+coef*(perty(i,jj)*(c(ww))*(tny+d(ww))))))/(aka)
end
if pert(i,jj)>perty(i,ww) then
    soma=soma+abs(((coef*(perty(i,ww)*(a(jj))*(tn+b(jj)))+coef*(perty(i,ww)*(c(ww))*(tny+d(ww))))))/(aka)
end
if pert(i,jj)==perty(i,ww) then
    soma=soma+abs(((coef*(perty(i,ww)*(a(jj))*(tn+b(jj)))+coef*(perty(i,ww)*(c(ww))*(tny+d(ww))))))/(aka)
end
end

// nova proposta de inferencia
//if pert(i,jj)<perty(i,ww) then
// soma=soma+(a(jj)+b(jj)*tn)*perty(i,ww)
//end
//if perty(i,ww)<pert(i,jj) then
// soma=soma+(c(ww)+d(ww)*tny)*perty(i,jj)
//end
//if perty(i,ww)==pert(i,jj) then
// soma=soma+(c(ww)+d(ww)*tny)*perty(i,jj)
//end

end
end
//soma=maiorparcela
zfuzzy=soma
endfunction
//definicao da funcao erro quadratico a ser minimizada
// Aqui dentro deve fazer uma chamada ao Modelo Fuzzy Takagi Sugeno

```

```

function jc=custo(x, y, a, b, c, d, xxi, yyi, zi, fex, fey)
    si=size(x);siy=size(y);
    somaq=0;i=1
    jj=1
    for ii=1:tamanho

        zestimado=fuzzyTS(8,x(ii),y(ii),a,b,c,d,xxi,yyi,zi,fex,fey);
        erroq=(abs(zi(ii)-zestimado))
        somaq=somaq+erroq
    end
    // jc=(somaq/tamanho)^0.5 //adaptado para usar RMSE
    jc=somaq
endfunction

// A funcao GRAFICA() serve para o acompanhamento do treinamento
// através de gráficos
function graff=grafica()
    //chamada do melhor modelo para graficar
    testaz2=zeros(tamanho)
    // Chamada do Modelo Fuzzy TS para graficar
    aaa=ones(8);bbb=ones(8);ccc=ones(8);ddd=ones(8);xuf=ones(3);yuf=ones(3)

    aaa(1)=melhorposicaodetodos(1);
    aaa(2)=melhorposicaodetodos(2);
    aaa(3)=melhorposicaodetodos(3);
    aaa(4)=melhorposicaodetodos(4);
    aaa(5)=melhorposicaodetodos(5);
    aaa(6)=melhorposicaodetodos(6);
    aaa(7)=melhorposicaodetodos(7);
    aaa(8)=melhorposicaodetodos(8);

    bbb(1)=melhorposicaodetodos(9);
    bbb(2)=melhorposicaodetodos(10);
    bbb(3)=melhorposicaodetodos(11);
    bbb(4)=melhorposicaodetodos(12);
    bbb(5)=melhorposicaodetodos(13);
    bbb(6)=melhorposicaodetodos(14);
    bbb(7)=melhorposicaodetodos(15);
    bbb(8)=melhorposicaodetodos(16);

    //ab=0;
    //for i=1:8
    //ab=ab+aaa(i)+bbb(i)
    //end

    ccc(1)=melhorposicaodetodos(17);
    ccc(2)=melhorposicaodetodos(18);
    ccc(3)=melhorposicaodetodos(19);
    ccc(4)=melhorposicaodetodos(20);
    ccc(5)=melhorposicaodetodos(21);
    ccc(6)=melhorposicaodetodos(22);
    ccc(7)=melhorposicaodetodos(23);
    ccc(8)=melhorposicaodetodos(24);

    ddd(1)=melhorposicaodetodos(25);
    ddd(2)=melhorposicaodetodos(26);
    ddd(3)=melhorposicaodetodos(27);
    ddd(4)=melhorposicaodetodos(28);
    ddd(5)=melhorposicaodetodos(29);
    ddd(6)=melhorposicaodetodos(30);

```

```

ddd(7)=melhorposicaodetodos(31);
ddd(8)=melhorposicaodetodos(32);

//cdd=0;
//for i=1:8
//cdd=cdd+ccc(i)+ddd(i)
//end

xuf(1)=melhorposicaodetodos(33);
xuf(2)=melhorposicaodetodos(34);
xuf(3)=melhorposicaodetodos(35);
yuf(1)=melhorposicaodetodos(36);
yuf(2)=melhorposicaodetodos(37);
yuf(3)=melhorposicaodetodos(38);

//xxu=xuf(1)+xuf(2)+xuf(3)
//yyu=yuf(1)+yuf(2)+yuf(3)

wxi=xi;
wyi=yi;
for wwy=1:tamanho
testaz2(wwy)=fuzzyTS(8,wxi(wwy),wyi(wwy),aaa,bbb,ccc,ddd,xxi,yyi,zi,xuf,yuf)
end
clf()
subplot(221)
plot(zi)
plot(testaz2)
//plot(zi,zi)
subplot(222)
//plot(erres)
plot(seq);
graff=1;

subplot(223)
//plot(erres)
plot(ab,cdd);

subplot(224)
//plot(erres)
plot(xxu,yyu);

endfunction

// fim das funcoes

//inicio do programa PRINCIPAL

tamanho=8; //numero de pontos do treinamento
yi=zeros(tamanho);xi=zeros(tamanho);zi=zeros(tamanho)
zinormalizado=ones(tamanho);xinormalizado=ones(tamanho);yinormalizado=ones(tamanho)

// dados para pares 45 e 90 graus
//xi=[5.919;6.045;7.127;11.934;10.994;13.089;13.069;19.997]; //DADOS DE TREINAMENTO para 45 graus
//yi=[7.757;13.078;14.017;18.012;18.941;21.989;19.038;21.130]; //DADOS DE TREINAMENTO para 90
graus
//zi=[0.228;0.527;0.726;0.967;1.203;1.440;1.950;2.48]; //DADOS DE TREINAMENTO de concentracao de
biomassa

// dados para pares 45 e 90 graus - a partir do segundo
//xi=[6.045;7.127;11.934;10.994;13.089]; //DADOS DE TREINAMENTO para 45 graus

```

```

//yi=[13.078;14.017;18.012;18.941;21.989]//;19.038;21.130]; //DADOS DE TREINAMENTO para 90 graus
//zi=[0.527;0.726;0.967;1.203;1.440]//;1.950;2.48]; //DADOS DE TREINAMENTO de concentracao de
biomassa

//xi=[6.045;7.127;11.934;10.994;13.089;13.069]; //DADOS DE TREINAMENTO para 45 graus - 6 dados
//yi=[13.078;14.017;18.012;18.941;21.989;19.038]; //DADOS DE TREINAMENTO para 90 graus - 6 dados
//zi=[0.527;0.726;0.967;1.203;1.440;1.950]; //DADOS DE TREINAMENTO de concentracao de biomassa -
6 dados

// dados para pares 45 graus e 135 graus
//xi=[5.919;6.045;7.127;11.934;10.994;13.089;13.069;19.997]; //DADOS DE TREINAMENTO para 45 graus
//yi=[50.742;69.062;59.220;68.165;47.847;41.874;27.11;20.142]; //DADOS DE TREINAMENTO para 135
graus
//zi=[0.228;0.527;0.726;0.967;1.203;1.440;1.950;2.48]; //DADOS DE TREINAMENTO de concentracao de
biomassa

// segundo ponto em diante
//xi=[6.045;7.127;11.934;10.994;13.089];//13.069;19.997]; //DADOS DE TREINAMENTO para 45 graus
//yi=[69.062;59.220;68.165;47.847;41.874]; //DADOS DE TREINAMENTO para 135 graus
//zi=[0.527;0.726;0.967;1.203;1.440]//;1.950;2.48]; //DADOS DE TREINAMENTO de concentracao de
biomassa

// dados para pares 90 graus e 135 graus
xi=[50.742;69.062;59.220;68.165;47.847;41.874;27.11;20.142]; //DADOS DE TREINAMENTO para 135 graus
yi=[7.757;13.078;14.017;18.012;18.941;21.989;19.038;21.130]; //DADOS DE TREINAMENTO para 90 graus
zi=[0.228;0.527;0.726;0.967;1.203;1.440;1.950;2.48]; //DADOS DE TREINAMENTO de concentracao de
biomassa

// dados para pares 90 graus e 135 graus - A partir do segundo
//xi=[69.062;59.220;68.165;47.847;41.874]//;27.11;20.142]; //DADOS DE TREINAMENTO para 135 graus
//yi=[13.078;14.017;18.012;18.941;21.989]//;19.038;21.130]; //DADOS DE TREINAMENTO para 90 graus
//zi=[0.527;0.726;0.967;1.203;1.440]//;1.950;2.48]; //DADOS DE TREINAMENTO de concentracao de
biomassa

//normalizacao dos dados de treinamento
maiorx=max(xi);maiory=max(yi);menorz=min(zi);maiorz=max(zi);
zinormalizado=zi./maiorz;zi=zinormalizado;
xinormalizado=xi./maiorx;xi=xinormalizado;
yinormalizado=yi./maiory;yi=yinormalizado;
// proposta de fuzificação ()
//dimensionamento de variáveis parâmetros do modelo
a=ones(8);b=ones(8);c=ones(8);d=ones(8);fex=ones(3);fey=ones(8)
//for ii=1:8
// a(ii)=rand();b(ii)=rand();
// c(ii)=rand();d(ii)=rand();

//end
espacofuzzyx=ones(tpop,3)
espacofuzzyy=ones(tpop,3)

for i=1:tpop
// geração das funções de pertinência da entrada
xxi=ones(3);xxi(1)=rand();xxi(2)=xxi(1)/10
while xxi(2)<=xxi(1)
xxi(2)=rand()
end
xxi(3)=xxi(1)/10
while xxi(3)<=xxi(2)
xxi(3)=rand()
end

```

```

espacofuzzyx(i,1)=xxi(1);
espacofuzzyx(i,2)=xxi(2);
espacofuzzyx(i,3)=xxi(3);
end

espacofuzzyx(tpop,1)=0.25;espacofuzzyx(tpop,2)=0.5;espacofuzzyx(tpop,3)=0.75;
espacofuzzyx(tpop-1,1)=0.20;espacofuzzyx(tpop-1,2)=0.4;espacofuzzyx(tpop-1,3)=0.6;
espacofuzzyx(tpop-2,1)=0.40;espacofuzzyx(tpop-2,2)=0.6;espacofuzzyx(tpop-2,3)=0.8;
espacofuzzyx(tpop-3,1)=0.33;espacofuzzyx(tpop-3,2)=0.66;espacofuzzyx(tpop-3,3)=0.99;
espacofuzzyx(tpop-4,1)=0.99;espacofuzzyx(tpop-4,2)=0.66;espacofuzzyx(tpop-4,3)=0.33;

for i=1:tpop
// geração das condicoes iniciais das funções de pertinência da entrada
yyi=ones(3);yyi(1)=rand();yyi(2)=yyi(1)/10
while yyi(2)<=yyi(1)
    yyi(2)=rand()
end
yyi(3)=yyi(1)/10
while yyi(3)<=yyi(2)
    yyi(3)=rand()
end
espacofuzzyy(i,1)=yyi(1);
espacofuzzyy(i,2)=yyi(2);
espacofuzzyy(i,3)=yyi(3);

espacofuzzyy(tpop,1)=0.25;espacofuzzyy(tpop,2)=0.5;espacofuzzyy(tpop,3)=0.75;
espacofuzzyy(tpop-1,1)=0.20;espacofuzzyy(tpop-1,2)=0.4;espacofuzzyy(tpop-1,3)=0.6;
espacofuzzyy(tpop-2,1)=0.40;espacofuzzyy(tpop-2,2)=0.6;espacofuzzyy(tpop-2,3)=0.8;
espacofuzzyy(tpop-3,1)=0.33;espacofuzzyy(tpop-3,2)=0.66;espacofuzzyy(tpop-3,3)=0.99;
espacofuzzyy(tpop-4,1)=0.99;espacofuzzyy(tpop-4,2)=0.66;espacofuzzyy(tpop-4,3)=0.33;

end
////
// inicializacao de parâmetros do PSO e auxiliares
gbest=1e9;igbest=0;imenor=0;vmax=0.7;c2=2;c1=2; ii=1
xgbest=0;ygbest=0;iteracoes=1; total=0
xygbest=0;
// definir precisao e comecar iteração

// dimensionamento das variáveis do PSO
velocidades=ones(tpop,38);
inercia=rand(tpop,1);

posicao=ones(tpop,38);
fitness=ones(tpop);
pbest=ones(tpop);
for k=1:tpop
pbest(k)=1e300;
end

melhorposicao=ones(tpop,38);
melhorposicaodetodos=ones(38);
// gera populacao aleatoria de a,b, c e d em pop0
// para as variaveis 33,34,35,36,37 e 38 (espacos fuzzy) as posicoes serão substituidas
// e as velocidades reduzidas
for i=1:tpop
    for j=1:38
        velocidades(i,j)=(0.05-rand()/10);
        posicao(i,j)=rand();
    end
end

```

```

end

for i=1:tpop
    for j=33:35
        velocidades(i,j)=(0.05-rand()/10);
    end
    posicao(i,33)=espacofuzzyx(i,1);
    posicao(i,34)=espacofuzzyx(i,2);
    posicao(i,35)=espacofuzzyx(i,3);

    for j=35:38
        velocidades(i,j)=(0.05-rand()/10);
    end

    posicao(i,36)=espacofuzzyy(i,1);
    posicao(i,37)=espacofuzzyy(i,2);
    posicao(i,38)=espacofuzzyy(i,3);

end
vezes=1
//começo das iterações

gbest=1e300;
fim=0

anterior=0

atual=gbest

//etapa de minizacao do indice de desempenho
while fim==0
    //inerzia=rand(tpop,1);
    for i=1:tpop
        for kkk=1:8
            a(kkk)=posicao(i,kkk);
            b(kkk)=posicao(i,kkk+8);
            c(kkk)=posicao(i,kkk+16);
            d(kkk)=posicao(i,kkk+24)
        end

        a(1)=posicao(i,1);
        a(2)=posicao(i,2);
        a(3)=posicao(i,3);
        a(4)=posicao(i,4);
        a(5)=posicao(i,5);
        a(6)=posicao(i,6);
        a(7)=posicao(i,7);
        a(8)=posicao(i,8);

        b(1)=posicao(i,9);
        b(2)=posicao(i,10);
        b(3)=posicao(i,11);
        b(4)=posicao(i,12);
        b(5)=posicao(i,13);
        b(6)=posicao(i,14);
        b(7)=posicao(i,15);
        b(8)=posicao(i,16);

        c(1)=posicao(i,17);
        c(2)=posicao(i,18);

```



```

c(3)=posicao(i,19);
c(4)=posicao(i,20);
c(5)=posicao(i,21);
c(6)=posicao(i,22);
c(7)=posicao(i,23);
c(8)=posicao(i,24);

d(1)=posicao(i,25);
d(2)=posicao(i,26);
d(3)=posicao(i,27);
d(4)=posicao(i,28);
d(5)=posicao(i,29);
d(6)=posicao(i,30);
d(7)=posicao(i,31);
d(8)=posicao(i,32);

fex(1)=espacofuzzyx(i,1);fex(2)=espacofuzzyx(i,2);fex(3)=espacofuzzyx(i,3);
fey(1)=espacofuzzyy(i,1);fey(2)=espacofuzzyy(i,2);fey(3)=espacofuzzyy(i,3);

fitness(i)=custo(xi,yi,a,b,c,d,xxi,yyi,zi,fex,fey);

    if fitness(i)<=pbest(i) then
        pbest(i)=fitness(i)
        for kk=1:38
            melhorposicao(i,kk)=posicao(i,kk)
        end
    //    pbestindex=i
    end

if pbest(i)<=gbest then
    melhorindex=i;
    gbest=pbest(i)
    for kk=1:38
        melhorposicaodetodos(kk)=posicao(i,kk)
    end
    amos=0;
    for kk=1:16
        amos=amos+posicao(i,kk)
    end
    ab(vezes)=amos;amos=0;
    for kk=17:32
        amos=amos+posicao(i,kk)
    end
    cdd(vezes)=amos;amos=0;
    xxu(vezes)=posicao(i,33)+posicao(i,34)+posicao(i,35)
    yyu(vezes)=posicao(i,36)+posicao(i,37)+posicao(i,38)

end
end
// calcula velocidade

for i=1:tpop
    for j=1:38
        // metaheurística da otimização do ALgoritmo PSO
        velocidades(i,j)=(inercia(i)*velocidades(i,j) + (c1 *(rand()) * (melhorposicao(i,j)-posicao(i,j)))+(c2*rand()
        *(melhorposicaodetodos(j)-posicao(i,j))))/(divisor)
        //velocidades(i,j)=velocidades(i,j)/(vezes+1);

    //sinal=0

```

```

//sorte=rand()
//if sorte>(1-vezes/10010) then
//if velocidades(i,j)<0 then
//sinal=-1
//end
//if velocidades(i,j)>0 then
//  sinal=1
//end
//if abs(velocidades(i,j))>0.3 then
//  velocidades(i,j)=sinal*0.3
//end
//end

posicao(i,j)=posicao(i,j)+velocidades(i,j)
//inercia variavel

inercia(i)=inercia(i)-(0.001*inercia(i));
if inercia(i)<0.02 then
  inercia(i)=rand()
end

//salvuarda do espaco fuzzy das funcoes de pertinencia
negativou=0;
jjj=33

while jjj<39
if posicao(i,jjj)<0 then
  negativou=1;
end
jjj=jjj+1;
end

if negativou==1 then

yyi=ones(3);yyi(1)=rand();yyi(2)=yyi(1)/10
while yyi(2)<=yyi(1)
  yyi(2)=rand()
end
yyi(3)=yyi(2)/10
while yyi(3)<=yyi(2)
  yyi(3)=rand()
end
posicao(i,33)=yyi(1);
posicao(i,34)=yyi(2);
posicao(i,35)=yyi(3);

yyi=ones(3);yyi(1)=rand();yyi(2)=yyi(1)/10
while yyi(2)<=yyi(1)
  yyi(2)=rand()
end
yyi(3)=yyi(2)/10
while yyi(3)<=yyi(2)
  yyi(3)=rand()
end
posicao(i,36)=yyi(1);
posicao(i,37)=yyi(2);
posicao(i,38)=yyi(3);

```

```

end

//Salvuarda
if posicao(i,34)<posicao(i,33) then
  while posicao(i,34)<posicao(i,33)
    posicao(i,33)=rand();
    posicao(i,34)=rand();
  end
end
if posicao(i,35)<posicao(i,34) then
  while posicao(i,35)<posicao(i,34)
    posicao(i,34)=rand();
    posicao(i,35)=rand();
  end
end
if posicao(i,35)>1 then
  while posicao(i,35)>1 & posicao(i,35)<posicao(i,34) then
    posicao(i,35)=rand();
  end
end

if posicao(i,36)<posicao(i,37) then
  while posicao(i,36)<posicao(i,37)
    posicao(i,36)=rand();
    posicao(i,37)=rand();
  end
end
if posicao(i,38)<posicao(i,37) then

  while posicao(i,38)<posicao(i,37)
    posicao(i,37)=rand();
    posicao(i,38)=rand();
  end
end
if posicao(i,38)>1 then
  while posicao(i,38)>1 & posicao(i,38)<posicao(i,37)
    posicao(i,38)=rand();
  end
end

maiorqueum=0
for kk=33:38
  if posicao(i,kk)>1.0 then
    maiorqueum=posicao(i,kk);
  end
end
if maiorqueum>0 then
  for kk=33:38
    posicao(i,kk)=posicao(i,kk)/maiorqueum;
  end
end
end

```

end

// fim salvaguarda

end

anterior=atual
atual=gbest

seq(vezes)=gbest
vezes=vezes+1
iteracoes=iteracoes+1

disp(gbest,vezes)

soma=0;
//inter2=0;

//reembaralha

if iteracoes>2 then
 inter2=inter2+1;
 iteracoes=1;

*// va=velmed*rand();*

mediass=zeros(33,1)
somass=zeros(338,1)

for jjjj=1:tpop
 for kkkk=1:38
 somass(kkkk)=somass(kkkk)+posicao(jjjj,kkkk)
 end
end

for kkkk=1:38
 mediass(kkkk)=somass(kkkk)/tpop;
end

for i=1:tpop
 if i<>melhorindex then
 for j=1:32

 velocidades(i,j)=(1-2*rand())/10;
 //posicao(i,j)=(mediass(j)+posicao(i,j))/2
 posicao(i,j)=mediass(j)+(0.05-rand()/10);;

 end
 end

//salvaguarda do espaco fuzzy

if posicao(i,34)<posicao(i,33) then

 while posicao(i,34)<posicao(i,33)
 posicao(i,33)=rand();
 posicao(i,34)=rand();

```

    end
end
if posicao(i,35)<posicao(i,34) then

    while posicao(i,35)<posicao(i,34)
        posicao(i,34)=rand();
        posicao(i,35)=rand();
    end
end

if posicao(i,36)<posicao(i,37) then

    while posicao(i,36)<posicao(i,37)
        posicao(i,36)=rand();
        posicao(i,37)=rand();
    end
end
if posicao(i,38)<posicao(i,37) then

    while posicao(i,38)<posicao(i,37)
        posicao(i,37)=rand();
        posicao(i,38)=rand();
    end
end

end

if inter2>20 then
    if vezes>150 then
        divisor=1;
    end
    saida=grafica()
    inter2=1;
    iteracoes=1;
    for i=1:tpop
        if i<>melhorindex then
            for j=1:38
                velocidades(i,j)=(0.05-rand()/10);
                posicao(i,j)=10*rand();
            end
        end
    end
    //salv guarda do espaco fuzzy
    //salv guarda do espaco fuzzy
    if posicao(i,34)<posicao(i,33) then

        while posicao(i,34)<posicao(i,33)
            posicao(i,33)=rand();
            posicao(i,34)=rand();
        end
    end
    if posicao(i,35)<posicao(i,34) then

        while posicao(i,35)<posicao(i,34)
            posicao(i,34)=rand();
            posicao(i,35)=rand();
        end
    end
end

```

```

end
if posicao(i,35)>1 then
    while posicao(i,35)>1 & posicao(i,35)<posicao(i,34)
        posicao(i,35)=rand();
    end
end
if posicao(i,36)<posicao(i,37) then
    while posicao(i,36)<posicao(i,37)
        posicao(i,36)=rand();posicao(i,37)=rand();
    end
end
if posicao(i,38)<posicao(i,37) then
    while posicao(i,38)<posicao(i,37)
        posicao(i,37)=rand();
        posicao(i,38)=rand();
    end
end
if posicao(i,38)>1 then
    while posicao(i,38)>1 & posicao(i,38)<posicao(i,37)
        posicao(i,38)=rand();
    end
end
maiorqueum=0
for kk=33:38
    if posicao(i,kk)>1.0 then
        maiorqueum=posicao(i,kk);
    end
end
if maiorqueum>0 then
    for kk=33:38
        posicao(i,kk)=posicao(i,kk)/maiorqueum;
    end
end
end
end

// fim salvaguarda

//disp ("vezes")
//disp(vezes)

erro_q_graf(vezes)=gbest
if vezes>1000 then
    fim=1
end

end

// Chamada do Modelo Fuzzy TS para graficar
aaa=ones(8);bbb=ones(8)
ccc=ones(8);ddd=ones(8)
xuf=ones(3);yuf=ones(3)

aaa(1)=melhorposicao(1);
aaa(2)=melhorposicao(2);
aaa(3)=melhorposicao(3);
aaa(4)=melhorposicao(4);

```

```

aaa(5)=melhorposicaodetodos(5);
aaa(6)=melhorposicaodetodos(6);
aaa(7)=melhorposicaodetodos(7);
aaa(8)=melhorposicaodetodos(8);

```

```

bbb(1)=melhorposicaodetodos(9);
bbb(2)=melhorposicaodetodos(10);
bbb(3)=melhorposicaodetodos(11);
bbb(4)=melhorposicaodetodos(12);
bbb(5)=melhorposicaodetodos(13);
bbb(6)=melhorposicaodetodos(14);
bbb(7)=melhorposicaodetodos(15);
bbb(8)=melhorposicaodetodos(16);

```

```

ccc(1)=melhorposicaodetodos(17);
ccc(2)=melhorposicaodetodos(18);
ccc(3)=melhorposicaodetodos(19);
ccc(4)=melhorposicaodetodos(20);
ccc(5)=melhorposicaodetodos(21);
ccc(6)=melhorposicaodetodos(22);
ccc(7)=melhorposicaodetodos(23);
ccc(8)=melhorposicaodetodos(24);

```

```

ddd(1)=melhorposicaodetodos(25);
ddd(2)=melhorposicaodetodos(26);
ddd(3)=melhorposicaodetodos(27);
ddd(4)=melhorposicaodetodos(28);
ddd(5)=melhorposicaodetodos(29);
ddd(6)=melhorposicaodetodos(30);
ddd(7)=melhorposicaodetodos(31);
ddd(8)=melhorposicaodetodos(32);

```

```

xuf(1)=melhorposicaodetodos(33);
xuf(2)=melhorposicaodetodos(34);
xuf(3)=melhorposicaodetodos(35);
yuf(1)=melhorposicaodetodos(36);
yuf(2)=melhorposicaodetodos(37);
yuf(3)=melhorposicaodetodos(38);

```

```
//for iii=1:8  SUSPEITO DE ERRO
```

```
//  aaa(iii)=melhorposicaodetodos(iii);bbb(iii)=melhorposicaodetodos(iii+8)
```

```
//  ccc(iii)=melhorposicaodetodos(iii+16);ddd(iii)=melhorposicaodetodos(iii+24)
```

```
//end
```

```
//for iii=1:3
```

```
//  xuf(iii)=melhorposicaodetodos(iii+32)
```

```
//  yuf(iii)=melhorposicaodetodos(iii+35)
```

```
//end
```

```
wxi=xi;
```

```
wyi=yi;
```

```
testaz=ones(tamanho)
```

```
//for ww=1:tamanho
```

```
  for wwy=1:tamanho
```

```
// avaliação do modelo fuzzy TS
```

```
testaz(wwy)=fuzzyTS(8,wxi(wwy),wyi(wwy),aaa,bbb,ccc,ddd,xxi,yyi,zi,xuf,yuf)
```

```

// zestimado=fuzzyTS(8,x(ii),y(ii),a,b,c,d,xxi,yyi,zi,fex,fey); //AQUI

end
//parar=input('parei');
//clf();
//subplot(221)
//plot(testaz);plot(zi)

//subplot(223)
//plot(zi,testaz);
//plot(zi,zi);

for iii=1:tamanho
erroerro(iii)=100*(zi(iii)-testaz(iii))/zi(iii)
end

// Calcula Coeficiente de Determinação R^2

// COEFICIENTE DE PEARSON
// REFERENDADO EM 26/12/2011
xis=ones(tamanho)
yps=ones(tamanho)
for iii=1:tamanho
    xis(iii)=zi(iii)
    yps(iii)=testaz(iii)
end
somaxis=0
somyis=0
somapxy=0
for iii=1:tamanho
    somaxis=somaxis+xis(iii)
    somyis=somyis+yps(iii)
end
mediax=somaxis/tamanho
mediay=somyis/tamanho
//calculo do numerador (COVARIANCIA DE X,Y)
for iii=1:tamanho
    ah=xis(iii)-mediax;
    bh=yps(iii)-mediay;
    produtoh=ah*bh;
    somapxy=somapxy+produtoh;
end
soma2=0;somab2=0
for iii=1:tamanho
    a2=(xis(iii)-mediax)^2;
    b2=(yps(iii)-mediay)^2;
    somaa2=somaa2+a2;
    somab2=somab2+b2;
end
raizvariânciax=somaa2^0.5;raizvariânciay=somab2^0.5;
// somapxy = covariância de X e Y;
// raizvariânciax = desvio padrao da var x
// raizvariânciay = desvio padrao da var y
coeficpearson=somapxy/(raizvariânciax*raizvariânciay);
//disp (xis,yps)
disp PEARSON
disp (coeficpearson)
disp PEARSON
erres(vezess)=coeficpearson;

```



```
vezess=vezess+1;

//subplot(222)
//plot(erres);

//if coeficpearson>0.99 then
// fim=1;
//end
end

//fim do calculo do coeficiente de determinacao
disp ('fim')

//FIM DO CODIGO
```

APÊNDICE E - Programa de Modelagem Cinética por VERHULST usando PSO para Arduino

```

// por RDPF
// junho de 2014
#include <math.h>
//*****
// Modelamento Cinético pelo uso da função Logística de Crescimento
// usando PSO para a minimização do somatório do erro quadrático
// por Renato Dutra Pereira Filho em 01/07/2014
unsigned long time1,time2;
void setup() {
  Serial.begin(9600);
}
void loop() {
float dados1[24],dados2[24], dados3[24];
float mpt1,mpt2,mpt3,mpt4;
int i;
float somax=0,somay=0,somax2=0,somaxy=0, num1, den1,coeficientelinear,m,comp,ymed,rquad;
int final;
float num2,den2, tangentedareta, tzero,f;

// ACRESCIDAS
int tpop=5; // número de partículas da populacao
float c1=2.;// fator de aprendizado cognitivo, normalmente 2
float c2=2.; // fator de aprendizado social, normalmente 2
float inercia=0.9; //valoracao do conhecimento ate a iteracao anterior
float vmax=0.2; //limitacao da velocidade das particulas
float vetor1[tpop],vetor2[tpop],vetor3[tpop],vetor4[tpop];
float maxiteracoes=1500.;
// dimensionamento das variáveis do PSO
// velocidade da partícula
// posicao da partícula
// fitness
float velocidades1[tpop],velocidades2[tpop],velocidades3[tpop],velocidades4[tpop];

```

```

float posicao1[tpop],posicao2[tpop],posicao3[tpop],posicao4[tpop];
float fitness[tpop];

//REACRESCIDAS
int vezes=1; // numero de épocas de treinamento
int vezes1=1;
float gbest=1e10; // gbest melhor resposta global
float pbest[tpop]; // pbest é a melhor resposta individual
float melhorposicaodetodos[4]; //armazena parametros de ajuste para gbest
float melhorposicao1[tpop],melhorposicao2[tpop],melhorposicao3[tpop],melhorposicao4[tpop]; //armazena
parâmetros de ajuste para a pop
int fim=0;
float anterior=0;
float atual=gbest;
time1=millis();
while (i<24){
    dados1[i]=0.0;
    dados2[i]=0.0;
    dados3[i]=0.0;
    i++;
}
dados1[1]=0.041666667; dados2[1]=0.156028902;
dados1[2]=0.083333333; dados2[2]=0.203131989;
dados1[3]=0.125; dados2[3]=0.233444435;
dados1[4]=0.166666667; dados2[4]=0.280147003;
dados1[5]=0.208333333; dados2[5]=0.297781238;
dados1[6]=0.25; dados2[6]=0.332736473;
dados1[7]=0.291666667; dados2[7]=0.355702608;
dados1[8]=0.333333333; dados2[8]=0.393771896;
dados1[9]=0.375; dados2[9]=0.420826833;
dados1[10]=0.416666667; dados2[10]=0.414246789;
dados1[11]=0.458333333; dados2[11]=0.41375507;
dados1[12]=0.5; dados2[12]=0.408968891;

```

```
dados1[13]=0.541666667; dados2[13]=0.430128455;
dados1[14]=0.583333333; dados2[14]=0.464023293;
dados1[15]=0.625;      dados2[15]=0.514578231;
dados1[16]=0.666666667; dados2[16]=0.508863291;
dados1[17]=0.708333333; dados2[17]=0.641047479;
dados1[18]=0.75;      dados2[18]=0.611256734;
dados1[19]=0.791666667; dados2[19]=0.602000101;
dados1[20]=0.833333333; dados2[20]=0.635247687;
dados1[21]=0.875;      dados2[21]=0.731158606;
dados1[22]=0.916666667; dados2[22]=0.904831698;
dados1[23]=0.958333333; dados2[23]=0.725323944;
dados1[24]=1.;      dados2[24]=0.850227028;
final=24;

// Definicao dos parâmetros do PSO
//Serial.println("DENTRO DO CODIGO");

// gera populacao aleatoria dos parametros a serem ajustados
for (int jj=1;jj<(tpop+1);jj++){
    velocidades1[jj]=aleav();
    velocidades2[jj]=aleav();
    velocidades3[jj]=aleav();
    velocidades4[jj]=aleav();
    posicao1[jj]=aleap();
    posicao2[jj]=aleap();
    posicao3[jj]=aleap();
    posicao4[jj]=aleap();
}

for (int iii=1;iii<(tpop+1);iii++){
    pbest[iii]=1e8;
}

//começo das iterações
fim=0;
```



```

//A funcao abaixo modela e determina o somatório do erro quadrático
// como funcao de
// xis e yis => dados de entrada de t e X (normalizados)
// vetor contem os parametros de ajuste, de acordo com a cinetica
// desejada

int tamanho=final;

float somaerroq=0;

float temp=0.0;

for (int kk=1;kk<(tamanho+1);kk++){

    temp=vetor1[jj]/(vetor2[jj]+vetor3[jj]*exp(-vetor4[jj]*dados1[kk]));

    //temp=vetor1[jj]/(1+vetor3[jj]*exp(-vetor4[jj]*dados1[kk]));

    somaerroq=somaerroq+(dados2[kk]-temp)*(dados2[kk]-temp);

}

//chamada da funcao de fitness //*****

fitness[jj]=somaerroq;

// Serial.println("");
// Serial.println("");
// Serial.println("");
// Serial.println(somaerroq);
// Serial.println("");
// Serial.println("");
// Serial.println("");
// Serial.println("");
// delay(5000);
// cc=fitness(i)

// avalia posicao de cada particula da populacao
if (fitness[jj]<=pbest[jj]) {
    pbest[jj]=fitness[jj];
}

// for (int jjj=1;jjj<(4+1);jjj++){

    melhorposicao1[jj]=posicao1[jj];

    melhorposicao2[jj]=posicao2[jj];

```

```

        melhorposicao3[jj]=posicao3[jj];
        melhorposicao4[jj]=posicao4[jj];
//    }
    if (pbest[jj]<=gbest) {
        gbest=pbest[jj];
        int gbestindex=jj;
        melhorposicaodetodos[1]=posicao1[jj];
        melhorposicaodetodos[2]=posicao2[jj];
        melhorposicaodetodos[3]=posicao3[jj];
        melhorposicaodetodos[4]=posicao4[jj];
    }
}

//g_de_t(vezes)=gbest;
// atualiza velocidades e posicoes das particulas
//
mpt1=melhorposicaodetodos[1];
mpt2=melhorposicaodetodos[2];
mpt3=melhorposicaodetodos[3];
mpt4=melhorposicaodetodos[4];
//Serial.print("GBEST = ");
//Serial.println(gbest);
}
//Serial.println("PBESTs E GBEST DETERMINADAS");
//LOOP DE ATUALIZACAO DA VELOCIDADE E DA POSICAO
for (i=1;i<(tpop+1);i++){
    // metaheurística da otimização Segundo Kennedy e Eberhart com inercia (SHI)
    //
    velocidades1[i]=inercia*velocidades1[i] + (c1 * alea() * (melhorposicao1[i]-posicao1[i])+c2*alea() *(mpt1-
posicao1[i]));
    velocidades2[i]=inercia*velocidades2[i] + (c1 * alea() * (melhorposicao2[i]-posicao2[i])+c2* alea() *(mpt2-
posicao2[i]));
    velocidades3[i]=inercia*velocidades3[i] + (c1 * alea() * (melhorposicao3[i]-posicao3[i])+c2*alea() *(mpt3-
posicao3[i]));

```

```

velocidades4[i]=inercia*velocidades4[i] + (c1 * alea() * (melhorposicao4[i]-posicao4[i])+c2*alea() *(mpt4-
posicao4[i]));
/*
if (abs(velocidades1[i])>vmax){
    velocidades1[i]=vmax;
}
if (abs(velocidades2[i])>vmax){
    velocidades2[i]=vmax;
}
if (abs(velocidades3[i])>vmax){
    velocidades3[i]=vmax;
}

if (abs(velocidades4[i])>vmax){
    velocidades4[i]=vmax;
}
*/
posicao1[i]=posicao1[i]+velocidades1[i];
posicao2[i]=posicao2[i]+velocidades2[i];
posicao3[i]=posicao3[i]+velocidades3[i];
posicao4[i]=posicao4[i]+velocidades4[i];

if (inercia>0.01){
inercia=inercia-0.1*inercia;
}
if (inercia<0.01){
    inercia=0.9;
}
}
//Serial.println("VELOCIDADES E POSICOES ATUALIZADAS");

```



```

anterior=atual;

atual=gbest;

vezes=vezes+1;

//disp ("vezes")

///disp(vezes)

if (vezes>maxiteracoes){

    fim=1;

}

if ((vezes-vezes1)>23){

    Serial.print(vezes);

    Serial.print(",");

    Serial.println(gbest);

    vezes1=vezes1+23;

}

} //fim do while

//disp ("menor")

//disp(anterior)

//chamada da funcao para tabular resultados

float ycalculado[24];

//vetor1[1]=mpt1;

//vetor2[2]=mpt2;

//vetor3[3]=mpt3;

//vetor4[4]=mpt4;

Serial.println("          ");

Serial.println("          ");

Serial.println("  TABULANDO          ");

Serial.println("          ");

Serial.println("          ");

//chamada da funcao de fitness //*****

for (int kk=1;kk<(24+1);kk++){

    float temp=mpt1/(mpt2+mpt3*exp(-mpt4*dados1[kk]));

    //float temp=mpt1/(1+mpt3*exp(-mpt4*dados1[kk]));

    Serial.print(dados1[kk]);

```

```

        Serial.print(",      ");
        ycalculado[kk]=temp;
        Serial.print(dados2[kk]);
        Serial.print(",      ");
        Serial.println(ycalculado[kk]);
    }

```

```

Serial.println("fim modelagem");
Serial.println("calculando r quadrado");
//Serial.println("fim");
//Serial.println("fim");
//delay(600000);
// Calcula Coeficiente de Determinação R^2
// COEFICIENTE DE PEARSON
// REFERENDADO EM 26/12/2011
float somaxis=0;
float somayis=0;
float somapxy=0;
for (int ii=1;ii<(24+1);ii++){
    somaxis=somaxis+dados2[ii];
    somayis=somayis+ycalculado[ii];
}
float mediax=0.0;
float mediay=0.0;
mediax=somaxis/24;
mediay=somayis/24;
//calculo do numerador (COVARIANCIA DE X,Y)
//float somapxy=0.0;
for (int iii=1;iii<(24+1);iii++){
    float ah=dados2[iii]-mediax;
    float bh=ycalculado[iii]-mediay;
    float produtoh=ah*bh;
    somapxy=somapxy+produtoh;
}

```

```

}
float somaa2=0;
float somab2=0;
for (int iii=1;iii<(24+1);iii++){
    float a2=(dados2[iii]-mediax)*(dados2[iii]-mediax);
    float b2=(ycalculado[iii]-mediay)*(ycalculado[iii]-mediay);
    somaa2=somaa2+a2;
    somab2=somab2+b2;
}
float raizvarienciax=0.;
float raizvarienciay=0.;
float raizvarienciax1=log(somaa2)*0.5;
float raizvarienciay1=log(somab2)*0.5;
raizvarienciax=exp(raizvarienciax1);
raizvarienciay=exp(raizvarienciay1);

// somapxy = covariancia de X e Y;
// raizvarienciax = desvio padrao da var x
// raizvarienciax = desvio padrao da var y
float coeficpearson=somapxy/(raizvarienciax*raizvarienciay);
//disp (xis,ypts)
Serial.println("PEARSON");
Serial.println(coeficpearson);
Serial.println("PEARSON");
Serial.println("=====");
Serial.println("Parametros do Modelo Logistico:");
Serial.print("Parametro mpt1:  ");
Serial.println(mpt1);
Serial.print("Parametro mpt2:  ");
Serial.println(mpt2);
Serial.print("Parametro mpt3:  ");
Serial.println(mpt3);

```

```

Serial.print("Parametro mpt4:  ");
Serial.println(mpt4);
Serial.println("=====");
time2=millis();
Serial.println("Tempos de Execução:");
Serial.println("Inicio:");
Serial.println(time1);
Serial.println("Fim:");
Serial.println(time2);
unsigned long dif;
dif=time2-time1;
Serial.println("Diferenca:");
Serial.println(dif);
Serial.println("FIM");
Serial.println("Memoria Livre:");
Serial.println(freeRam());
Serial.println("FIM");
delay(600000);
}
//*****

float alea(){
// randomSeed(analogRead(random(0,6)));

float result;
long menor=0;
long maior=2047483647;
result = (float) random(menor,maior)/(maior-menor);
return result;
}

float aleap(){
// randomSeed(analogRead(random(0,6)));

float result;
long menor=0;

```

```
long maior=2047483647;

result = (float) random(menor,maior)/(maior-menor);

result=result*1e3;

return result;
}

float aleav(){
// randomSeed(analogRead(random(0,6)));

float result;

long menor=0;

long maior=2047483647;

result = (float) random(menor,maior)/(maior-menor);

result=(1-(0.5*result));

return result;
}

int freeRam () {

extern int __heap_start, *__brkval;

int v;

return (int) &v - (__brkval == 0 ? (int) &__heap_start : (int) __brkval);

}

// FIM DO CODIGO
```