UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

TIAGO MAGALHÃES SOARES

DESENVOLVIMENTO DE CONTROLE AVANÇADO PARA PROCESSOS DE
AERAÇÃO POR AR DIFUSO VISANDO APLICAÇÃO EM SISTEMAS DE
TRATAMENTO DE EFLUENTES

São Carlos

2010

TIAGO MAGALHÃES SOARES

DESENVOLVIMENTO DE CONTROLE AVANÇADO PARA PROCESSOS DE
AERAÇÃO POR AR DIFUSO VISANDO APLICAÇÃO EM SISTEMAS DE
TRATAMENTO DE EFLUENTES

Dissertação apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo como parte dos requisitos para obtenção do título de Mestre em Ciências (Engenharia Hidráulica e Saneamento).

Orientador: Prof. Dr. Nivaldo
Aparecido Corrêa

São Carlos

2010

FOLHA DE JULGAMENTO

Candidato(a): Engenheiro **TIAGO MAGALHÃES SOARES.**

Dissertação defendida e julgada em 10/05/2010 perante a Comissão Julgadora:

Prof. Dr. **NIVALDO APARECIDO CORRÊA - (Orientador)**
(Escola de Engenharia de São Carlos/USP) Aprovado

Prof. Associado **MARCELO ZAIAT**
(Escola de Engenharia de São Carlos/USP) Aprovado

Prof. Dr. **WU HONG KWONG**
(Universidade Federal de São Carlos/UFSCar) Aprovado

Prof. Associado **MARCELO ZAIAT**
Coordenador do Programa de Pós-Graduação em
Engenharia Hidráulica e Saneamento.

Prof. Associado **GERALDO ROBERTO MARTINS DA COSTA**
Presidente da Comissão da Pós-Graduação da EESC

À minha Mãe, Arlete

Pelo carinho e amor, mesmo tão distante.

AGRADECIMENTOS

À Escola de Engenharia de São Carlos (EESC), pela oportunidade de realização do curso de mestrado.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pela concessão da bolsa de mestrado.

Ao Departamento de Hidráulica e Saneamento (SHS) e ao Centro de Recursos Hídricos e Ecologia Aplicada (CRHEA), por toda estrutura disponibilizada.

Ao Professor Dr. Nivaldo Aparecido Corrêa, pela orientação, compreensão e oportunidades concedidas.

Aos Professores e aos funcionários do Departamento de Hidráulica e Saneamento (SHS), pela colaboração e apoio na realização do trabalho.

Aos mestres da Universidade Federal de Viçosa e da Universidade de São Paulo, pelos ensinamentos durante o curso de graduação em Engenharia Ambiental e de pós-graduação em Ciências (Engenharia Hidráulica e Saneamento).

Aos amigos criados em São Carlos e aqueles vindos de Viçosa, pela amizade, pelos momentos de descontração e por tornarem o período em São Carlos mais agradável.

Aos amigos que mesmo de longe estiveram sempre por perto.

À minha família: Arlete, José, Micheli, Joarle, Josarlete, Rodrigo e Heloísa, pelo amor, amizade e presença sempre tão marcante.

Aos meus avós, tios, tias, primos, primas, sobrinhos e afilhados.

À minha nova família, os familiares da Micheli.

A Micheli, pelo amor e carinho.

Enfim, a todos que de alguma maneira contribuíram para a realização e finalização deste trabalho.

“(...) seria mais fácil fazer como todo mundo faz
sem sair do sofá, deixar a ferrari pra trás
seria mais fácil, como todo mundo faz
o milésimo gol sentado na mesa de um bar

mas nós vibramos em outra frequência
sabemos que não é bem assim
se fosse fácil achar o caminho das pedras
tantas pedras no caminho não seria ruim (...)”

Humberto Gessinger

RESUMO

SOARES, T.M. **Desenvolvimento de Controle Avançado para Processos de Aeração por Ar Difuso Visando Aplicação em Sistemas de Tratamento de Efluentes**. 2010. 148p. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2010.

Sistemas de aeração são amplamente utilizados em estações de tratamento de esgotos sanitários e efluentes industriais, sendo a configuração mais utilizada os sistemas de lodos ativados. Neste trabalho, buscou-se a implementação e o teste de uma estratégia de controle avançado em um sistema de aeração. Foi utilizado um canal de aeração com recirculação de água como simulação de um tanque de aeração. Para consumir o oxigênio dissolvido na água, foi realizada a dosagem de hipossulfito de sódio na entrada de água do canal. Um algoritmo de controle preditivo tipo QDMC (*“Quadratic Dynamic Matrix Control”*) foi implementado. Para estabelecer o *“set-point”* de OD (oxigênio dissolvido) pela manipulação da rotação do soprador de ar, foi utilizado um microcomputador com programação desenvolvida em linguagem C. Foi instalado na saída do canal uma sonda com sensores de OD e temperatura e um inversor de frequência foi ligado ao soprador, ambos conectados ao microcomputador para armazenamento das informações geradas e envio dos comandos gerados pelo algoritmo de controle. Foram avaliadas a performance e estabilidade da estratégia de controle adotada, mostrando que o controlador desenvolvido é robusto e estável para utilização em sistemas de aeração. Isto indica que é interessante desenvolver o estudo para reatores biológicos operando em estações de tratamento de efluentes.

Palavras-chave: aeração, controle preditivo, QDMC, tratamento aeróbico de efluentes.

ABSTRACT

SOARES,T.M. Development of Advanced Control Process of Aeration Tank with Air Difusion for application in Wastewater Treatment Plants. 2010. 148p. M. Sc. Dissertation – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2010.

Aeration systems are widely used in wastewater treatment plants, and the configuration more common is activated sludge plant. In this study, we sought to implement and test an advanced control strategy in an aeration system. We used a channel aeration with recirculation of water as a simulation of an aeration tank. To consume the dissolved oxygen in water was performed dosage of sodium thiosulphate in water entering the channel. A predictive control algorithm type QDMC ("Quadratic Dynamic Matrix Control") was implemented. To establish the set-point of DO (dissolved oxygen) by manipulating the rotation of the air blower was used with a microcomputer program developed in C language. It was installed onto the end of channel a sensor of DO and temperature and a frequency converter was connected to the blower, both connected to the computer for storing information generated and transmission of information generated by the control algorithm. We evaluated the performance and stability of the control strategy adopted, showing that the developed controller is robust and stable for use in aeration systems. This indicates that it is interesting to further explore the biological reactors operating in sewage treatment effluents.

Keywords: aeration; predictive control; QDMC; wastewater treatment.

LISTA DE FIGURAS

Figura 1. Lodos ativados convencional: tanque de aeração seguido de decantador secundário.	38
Figura 2. Representação esquemática das etapas de onde ocorre a resistência ao transporte de oxigênio da bolha de ar às células microbianas. Fonte: adaptado de Bailey e Ollis, 1986.	47
Figura 3. Perfil 24 horas de vazão afluente a um tanque de aeração recebendo efluente de indústria de embalagens de papel. Fonte: BEAL et al., 2006.	54
Figura 4. Variação no tempo da entrada e da saída de processo (a) não-controlado e (b) controlado. Fonte: adaptado de Andrews, 1974.	55
Figura 5. Estrutura básica para o controle de oxigênio dissolvido em tanques de aeração de sistemas de tratamento por lodos ativados. Fonte: adaptado de Metcalf & Eddy, 2003.	59
Figura 6. Ações passadas, atual e o futuro predito do processo controlado. Fonte: adaptado de García et al. (1989).	62
Figura 7. Planta experimental do tanque de aeração, dimensões e disposição do sistema para aquisição de dados e controle.	67
Figura 8. Fotografia do aparato experimental instalado no laboratório, com destaque para a bomba de recirculação de água, o rotâmetro e a dosagem de hipossulfito de sódio.	68
Figura 9. Funcionamento do controlador aplicado ao canal de aeração em estudo. 73	
Figura 10. Resposta ao degrau na variável manipulada.	74
Figura 11. Interface gráfica da programação utilizada no experimento.	83
Figura 12. Visão lateral do canal de aeração em operação.	84
Figura 13. Visão da parte superficial da água no canal de aeração em operação. 84	

Figura 14. Resultado da perturbação degrau negativo no “ <i>set-point</i> ” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.....	85
Figura 15. Resultado da perturbação degrau negativo no “ <i>set-point</i> ” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.....	86
Figura 16. Resultado da perturbação degrau negativo no “ <i>set-point</i> ” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.....	86
Figura 17. Resultado da perturbação degrau negativo no “ <i>set-point</i> ” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.....	87
Figura 18. Resultado da perturbação degrau negativo no “ <i>set-point</i> ” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.....	87
Figura 19. Resultado da perturbação degrau negativo no “ <i>set-point</i> ” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.....	88
Figura 20. Resultado da perturbação degrau negativo no “ <i>set-point</i> ” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.....	88
Figura 21. Resultado da perturbação degrau positivo no “ <i>set-point</i> ” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.....	89
Figura 22. Resultado da perturbação degrau positivo no “ <i>set-point</i> ” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.....	89
Figura 23. Resultado da perturbação degrau positivo no “ <i>set-point</i> ” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.....	90

Figura 24. Resultado da perturbação degrau positivo no “ <i>set-point</i> ” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.....	90
Figura 25. Perturbação tipo degrau negativo no “ <i>set-point</i> ”; tempo de controle (k) de 100 segundos.	94
Figura 26. Perturbação tipo degrau positivo no “ <i>set-point</i> ”; tempo de controle (k) de 100 segundos.	95
Figura 27. Perturbação crítica no “ <i>set-point</i> ”: degrau negativo seguido de degrau positivo; tempo de controle (k) de 50 segundos.	97
Figura 28. Perturbação crítica no “ <i>set-point</i> ”: degrau positivo seguido de degrau negativo; tempo de controle (k) de 100 segundos.....	98
Figura 29. Perturbação crítica no “ <i>set-point</i> ”: degrau negativo seguido de degrau positivo; tempo de controle (k) de 150 segundos.	99
Figura 30. Perturbação crítica na vazão de água do canal de aeração; tempo de controle (k) de 50 segundos.	101
Figura 31. Perturbação crítica na vazão de água do canal de aeração; tempo de controle (k) de 100 segundos.	102
Figura 32. Perturbação crítica na vazão de água do canal de aeração; tempo de controle (k) de 150 segundos.	103
Figura 33. Operação em malha fechada sem perturbação premeditada; tempo de controle (k) de 100 segundos.	105

LISTA DE TABELAS

Tabela 1.	Características dos experimentos realizados para identificação do modelo de processo.....	71
Tabela 2.	Resultados dos testes de perturbação degrau no set-point para identificação do modelo de processo.....	91

LISTA DE SIGLAS E ABREVIATURAS

CRHEA	Centro de Recursos Hídricos e Ecologia Aplicada.
DBO _{5,20}	Demanda Bioquímica de Oxigênio de 5 dias a 20 °C.
DMC	“Dynamic Matrix Control” (Controle por Matriz Dinâmica).
DQO	Demanda Química de Oxigênio.
ETE	Estação de Tratamento de Efluentes.
ETEs	Estações de Tratamento de Efluentes.
MBC	“Model Based Control” (Controle Baseado em Modelo).
MIMO	“Multiple Input, Multiple Output” (Múltipla Entrada, Múltipla Saída).
MPC	“Model Predictive Control” (Controle Preditivo por Modelo).
NARMAX	“Non-Linear Auto-Regressive Moving Average with Exogeneous Input” (Movimento Médio Auto-Regressivo Não-Linear com Entrada Externa).
OD	Oxigênio Dissolvido.
pH	Potencial Hidrogeniônico.
PI	Proporcional Integrativo.
PID	Proporcional Integrativo Derivativo.
ppm	Partes por milhão.
QDMC	“Quadratic Dynamic Matrix Control” (Controle por Matriz Dinâmica Quadrática)
QP	“Quadratic Programing” (Programação Quadrática)
rot	Rotação.
rpm	Rotações por minuto.
SISO	“Simple Input, Simple Output” (Simples Entrada, Simples Saída).

LISTA DE SÍMBOLOS

A = amplitude do degrau na variável de entrada [rpm];

$\mathbf{A}$ = matriz dinâmica com dimensão $R \times L$;

A_i = área interfacial para a transferência de massa [mm^2];

C = concentração de oxigênio dissolvido [$mg \cdot L^{-1}$];

C_0 = concentração inicial de oxigênio dissolvido [$mg \cdot L^{-1}$];

C_I = concentração de oxigênio na interface líquida [$mg \cdot L^{-1}$];

C_L = concentração de oxigênio dissolvido no líquido [$mg \cdot L^{-1}$];

C_s = concentração de saturação de oxigênio dissolvido no líquido [$mg \cdot L^{-1}$];

C_{Si} = concentração de saturação do oxigênio na interface [$mg \cdot L^{-1}$];

C^* = concentração de equilíbrio para a pressão parcial do gás P_b ;

C^{**} = concentração instantânea de oxigênio dissolvido [$mg \cdot L^{-1}$];

D = difusividade molecular do oxigênio na água [$mg \cdot mm^{-2}$];

D_t = difusão turbulenta [$mm^2 \cdot s^{-1}$];

$\frac{dC}{dt}$ = taxa de transferência de oxigênio [$mg \cdot L^{-1} \cdot s^{-1}$];

$\hat{\mathbf{E}}'$ = vetor que representa o desvio entre o “set-point” e a predição da variável controlada;

H = constante de Henry [$atm \cdot m^{-3} \cdot mol^{-1}$];

h_i = valor do pulso no i -ésimo intervalo de tempo discreto;

K = ganho do processo [$\frac{ppm}{rpm}$];

k_G = coeficientes de transferência de massa para o filme gasoso [s^{-1}];

k_g = velocidade de transferência de massa no filme gasoso viscoso [$mm \cdot s^{-1}$];

k_L = coeficientes de transferência de massa para o filme líquido [s^{-1}];

k_l = velocidade de transferência de massa no filme líquido viscoso [$mm \cdot s^{-1}$];

K_L = coeficiente de transferência de massa global [$mm \cdot s^{-1}$];

k_{La} = coeficiente volumétrico global de transferência de oxigênio [s^{-1}];

N = número de termos da seqüência de resposta ao degrau ou horizonte do modelo;

M = massa de oxigênio transferido por unidade de tempo [$mg \cdot s^{-1}$];

O_2 = representação da molécula de oxigênio;

P_B = pressão parcial no meio gasoso [$kg \cdot cm^{-1} \cdot s^{-2}$];

P_I = pressão parcial na interface [$kg \cdot cm^{-1} \cdot s^{-2}$];

P^* = pressão de equilíbrio para a concentração do gás C_L ;

Q e R = matrizes de ponderação definida e positiva;

R = horizonte de predição;

r_m = velocidade da ação enzimática microbiana [$mg \cdot L^{-1} \cdot s^{-1}$];

$S_3O_4^{-2}$ = representação da molécula do íon hipossulfito;

SO_4^{-2} = representação da molécula do íon sulfato;

t = tempo [s];

t_c = tempo retenção celular [s];

u = variável manipulada;

u_{max} = limite máximo para a variável manipulado;

u_{min} = limite mínimo para a variável manipulada;

Y = resposta ao estímulo [ppm];

Y_i = estado estacionário inicial [ppm];

y_k = valor medido para a variável controlada no tempo k ;

$\hat{y}_k$ = valor predito para a variável controlada no tempo k ;

$\hat{y}_{k+1}$ = valor predito para a variável controlada no tempo $k+1$;

y_{k+j-1} = valor medido da variável controlada no tempo $k+j-1$;

y_{k+j}^c = predição corrigida da variável controlada no tempo $k+j$;

y_{k+j}^d = valor desejado para a variável controlada no próximo instante de tempo;

y_k^{sp} = “*set-point*” da variável manipulada;

z = distância da interface [mm];

α^j = velocidade de aproximação do “*set-point*”;

$\Delta \mathbf{u}$ = movimento de controle;

Δu_{k+1-i} = ação controladora no tempo $k+1-i$;

$\Delta \mathbf{u}_{\max}$ = movimento máximo de controle;

$\Delta \mathbf{u}_{\min}$ = movimento mínimo de controle;

τ = constante de tempo [s];

θ = tempo de atraso [s];

$(\mathbf{A}^T \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^T$ = matriz pseudo-inversa.

SUMÁRIO

1.	INTRODUÇÃO.....	37
1.1.	Estrutura da Dissertação.....	40
2.	OBJETIVO	43
2.1.	Objetivos Específicos.....	43
3.	PROCESSO DE AERAÇÃO EM ESTAÇÕES DE TRATAMENTO DE EFLUENTES	45
3.1.	Fundamentos da Transferência de Oxigênio	45
3.2.	Dificuldades Operacionais em Sistemas de Aeração.....	53
4.	CONTROLE DE PROCESSOS DE AERAÇÃO.....	55
4.1.	Estratégias de controle.....	56
4.2.	Estrutura básica para controle de oxigênio dissolvido em sistemas de aeração.....	58
5.	ALGORITMOS DE CONTROLE	61
6.	APLICAÇÕES.....	65
7.	MATERIAL E MÉTODOS	67
7.1.	Aparato Experimental.....	67
7.2.	Determinação da faixa operacional.....	69
7.3.	Identificação do modelo do processo de aeração	70
7.4.	Controlador	72
7.5.	Descrição matemática do algoritmo de controle	73
8.	RESULTADOS E DISCUSSÃO	83
8.1.	Identificando o modelo do processo de aeração.....	85
8.2.	Experimentos em malha fechada.....	92
9.	CONCLUSÕES.....	107

10. RECOMENDAÇÕES	109
11. REFERÊNCIAS BIBLIOGRÁFICAS	111
ANEXO I.....	117
Programação em linguagem C para operacionalização do controlador QDMC 117	

1. INTRODUÇÃO

Os sistemas de tratamento de águas residuárias possuem mecanismos e processos que tratam as águas contaminadas por indústrias ou qualquer atividade humana. As estações de tratamento de efluentes são projetadas visando tanto o lançamento dessas águas aos ambientes aquáticos quanto o reúso de água. Muitas são as fontes de águas residuárias, mas atualmente busca-se minimizar a produção de efluentes e fazer o reúso de águas nos processos de produção. Porém, a geração de efluentes é inevitável para a maioria das atividades e os sistemas de tratamento existirão sempre pensando na preservação da qualidade dos recursos hídricos.

A grande variedade das atividades faz com que cada tipologia industrial tenha efluentes com características próprias. Ainda assim, cada fábrica terá particularidades em termos quantitativos e, principalmente, qualitativos em seus efluentes. Ou seja, o projeto de um sistema de tratamento de efluentes deverá atender essas particularidades para que o tratamento de efluentes aconteça efetivamente e alcance o objetivo de preservar a qualidade dos corpos d'água onde os despejos industriais são lançados.

Uma Estação de Tratamento de Efluentes (ETE) poderá ser constituída por diversos níveis de tratamento, sendo que cada um desses níveis terá uma unidade projetada especificamente para tal finalidade.

Para o tratamento secundário existem algumas configurações clássicas, já consagradas pela teoria e pela prática do tratamento de efluentes. A principal delas são os sistemas de lodos ativados, devido a sua alta eficiência e pequenos requisitos de área construída. Sua concepção inicial foi focada na remoção de poluentes orgânicos biodegradáveis, mas na atualidade já existem diversas configurações abrangendo também a remoção de nitrogênio e fósforo.

Figura 1. Lodos ativados convencional: tanque de aeração seguido de decantador secundário.

O tanque de aeração é o reator biológico aeróbio, onde ocorrerá a remoção da matéria orgânica do efluente. A aeração é realizada por meio de aeradores de superfície ou difusores de ar e são necessários para aumentar a transferência de oxigênio do ar para os microrganismos, que oxidam matéria orgânica e inorgânica. É importante destacar que o suprimento de oxigênio deve ser equilibrado, ou seja, deve-se evitar baixas concentrações de oxigênio dissolvido e, ao mesmo tempo, excessos de aeração, sempre buscando a racionalização do consumo de energia.

O material orgânico agora se encontra na forma de biomassa em suspensão. O decantador secundário tem como função a remoção dessa biomassa presente no efluente. A biomassa concentrada no fundo do decantador, ou lodo secundário, é, em parte, retornada para o tanque de aeração. O lodo restante é encaminhado para tratamento e disposição final.

A primeira oportunidade para controle de sistemas de lodos ativados é na vazão de entrada. A vazão normalmente é controlada com a construção de tanques de equalização, principalmente em estações de tratamento de efluentes industriais. No caso de esgotos sanitários, a equalização pode inviabilizar o projeto devido às grandes oscilações diárias. Esse problema pode ser parcialmente resolvido com a construção de várias estações elevatórias dentro da rede de coleta de esgotos.

Em Quebec, Canadá, a estação de tratamento de esgotos sanitários municipal opera com sistema de controle de vazão em tempo real, com controlador preditivo operando na rede coletora de esgotos (SCHÜTZE, et al., 2004). O sistema de controle utiliza dados do monitoramento de vazão, de nível d'água na rede coletora, de intensidade das precipitações, de imagens de radar e de previsões de chuva em intervalos de 2 horas.

O controle dos reatores biológicos é fundamental para garantir eficiência de tratamento e redução de custos de operação em sistemas de lodos ativados. Estuda-se principalmente o controle da concentração de oxigênio dissolvido, devido à importância do fornecimento de oxigênio para o crescimento microbiano heterotrófico e porque os sistemas de aeração são os principais consumidores de energia. O controle de processos de aeração também é interessante em processos com remoção de nutrientes, pois estes sistemas exigem ambientes controlados e condições operacionais bem definidas.

Na separação secundária de sólidos também existem possibilidades para utilização do controle de processos. A elevação da manta de lodo e a taxa de recirculação de lodo são essenciais para o bom funcionamento do decantador secundário.

O foco do controle de processos em estações de tratamento de efluentes por lodos ativados é voltado principalmente para os sistemas de aeração. Isto ocorre principalmente devido à necessidade de redução de custos operacionais e à necessidade de sistemas de tratamento com alta eficiência de remoção de poluentes (HAMITLON et al., 2006), visto a excessiva demanda de energia dos aeradores (superficiais ou por ar difuso) e a exigência de qualidade cada vez maior para os efluentes tratados.

Os processos de tratamento de efluentes são como processos de produção industrial e, assim como estes, estão sujeitos a aplicação de controladores automáticos para melhoria de eficiência e redução de custos operacionais. Por isso, a implantação de controle de processos de aeração em ETEs é importante para adequado desempenho operacional, buscando sempre a eficiência energética e de qualidade do tratamento de efluentes.

Segundo Correa (2006), a performance do controle tipo PID é satisfatória para o sistema de aeração de água por ar difuso em canal e recomenda a implementação de controle avançado para este tipo de processo. Por isso, será dada sequência ao trabalho referenciado, conforme será apresentado neste texto.

1.1. Estrutura da Dissertação

O objetivo deste trabalho é desenvolver um controlador preditivo para o tanque de aeração de um sistema biológico de tratamento de efluentes, como apresentado no item 2. No item 3 – Processo de Aeração em Estações de Tratamento de Efluentes, são apresentados os fundamentos do processo de aeração, abordando os processos de transferência de massa e suas teorias, e as dificuldades encontradas na operação das estações de tratamento de efluentes, destacando as variações de carga de poluentes tanto devido a variações de concentração quanto a variações na vazão afluyente. No item 4 – Controle de Processos de Aeração, será possível conhecer as diversas estratégias de controle bem como a estrutura básica necessária para implementação de um sistema de controle de oxigênio dissolvido. O item 5 – Algoritmos de Controle aborda alguns dos possíveis formas de desenvolvimento do controlador, como controladores preditivos, adaptativos, etc. Já no item 6 – Aplicações é possível conhecer algumas das aplicações de controle de processos em sistemas de aeração, mostrando que o tema é importante e deve ser estudado nas suas diversas variáveis.

Passando para a parte do experimento desenvolvido, no item 7 – Material e Métodos é possível conhecer o aparato experimental utilizado no desenvolvimento do trabalho, como foi a identificação do modelo de processo, como foi determinada a faixa operacional do controlador, como foi desenvolvido a controlador e sua descrição matemática. No item 8 – Resultados e Discussão são apresentados e discutidos os dados gerados no experimento de identificação do modelo de processo e os resultados gerados na operação do sistema em malha fechada. Por fim, no item 9 – Recomendações e Conclusões são tiradas as conclusões que o trabalho desenvolvido permitiu e relatadas as recomendações para os próximos experimentos quando da continuidade de desenvolvimento deste estudo.

Finalmente, no item 10 – Referências Bibliográficas são apresentadas as obras consultadas para desenvolvimento deste trabalho. O Anexo I traz a programação em linguagem C utilizada para processamento do controlador no microcomputador.

2. OBJETIVO

O objetivo do projeto foi desenvolver um controlador avançado para o processo de aeração por ar difuso, em canal de circulação de água, com a finalidade de aplicação em estações de tratamento de efluentes que utilizam reatores aeróbios para remoção de poluentes nos processos biológicos.

2.1. Objetivos Específicos

Os objetivos específicos deste trabalho foram:

- Caracterizar o processo e determinar os parâmetros do modelo de primeira ordem representativos da aeração de água por ar difuso em canal com vistas à construção de controladores avançados para processos de aeração em meios líquidos;
- Implementar malha de controle avançado do tipo QDMC para sistemas de aeração por ar difuso em tanques/canais de aeração, mostrando a viabilidade de sua aplicação em reatores aeróbios de estações de tratamento de efluentes.

3. PROCESSO DE AERAÇÃO EM ESTAÇÕES DE TRATAMENTO DE EFLUENTES

3.1. Fundamentos da Transferência de Oxigênio

A reoxigenação da água, a partir do turbilhonamento ou da injeção de ar/oxigênio, é um processo bem conhecido. A variação da concentração de oxigênio dissolvido (OD) na água pode ser descrita através do balanço de massa:

$$\frac{dC}{dt} = k_{La} \cdot (C_s - C_L) \quad (1)$$

com a condição inicial:

$$t = 0 \rightarrow C_L = C_0$$

soluciona-se a Equação diferencial (1):

$$C_L = C_s + (C_0 - C_s) \cdot e^{-k_{La} \cdot t} \quad (2)$$

em que:

$$\frac{dC}{dt} = \text{taxa de transferência de oxigênio } [mg \cdot L^{-1} \cdot s^{-1}];$$

$$k_{La} = \text{coeficiente volumétrico global de transferência de oxigênio } [s^{-1}];$$

$$C_s = \text{concentração de saturação de oxigênio dissolvido no líquido } [mg \cdot L^{-1}];$$

$$C_0 = \text{concentração inicial de oxigênio dissolvido } [mg \cdot L^{-1}];$$

$$C_L = \text{concentração de oxigênio dissolvido no líquido } [mg \cdot L^{-1}].$$

No caso de sistemas aeróbios de tratamento de águas residuárias, estão presentes diversas substâncias, orgânicas e inorgânicas, e população estável

de microrganismos, que consomem oxigênio dissolvido para estabilização da matéria orgânica. Desta forma, o processo de reoxigenação da água terá comportamento diferenciado.

A determinação do coeficiente de transferência de oxigênio é, normalmente, realizada em relação à água pura. No caso de águas residuárias, a correção dos parâmetros cinéticos é importante para o dimensionamento e escolha de equipamentos de aeração. Sistemas mal dimensionados podem não suprir a necessidade de oxigênio dissolvido dos microrganismos, reduzindo a eficiência ou mesmo não funcionando como um sistema aeróbio. A correção dos parâmetros cinéticos seguirá o seguinte preceito:

$$\text{coeficiente de correção} = \frac{\text{parâmetro sob condições de processo}}{\text{parâmetro padrão}} \quad (3)$$

Considerando a presença de substâncias e microrganismos na água residuária em um reator aeróbio para tratamento de efluentes, com as devidas correções nos parâmetros cinéticos, Viessman e Hammer (1998) apresentam o balanço de massa para o oxigênio dissolvido:

$$\frac{dC}{dt} = k_{La} \cdot (C_s - C) - r_m \quad (4)$$

em que:

$$\frac{dC}{dt} = \text{velocidade de transferência do oxigênio } [mg \cdot L^{-1} \cdot s^{-1}];$$

$$k_{La} = \text{coeficiente volumétrico global de transferência de oxigênio } [s^{-1}];$$

$$r_m = \text{velocidade da ação enzimática microbiana } [mg \cdot L^{-1} \cdot s^{-1}].$$

Este balanço de massa tem como principal característica a consideração a respeito do consumo de oxigênio pela microbiota presente no reator biológico onde a água residuária se encontra.

A disponibilidade de oxigênio para os microrganismos em um reator aeróbio depende da solubilidade do oxigênio na água, da velocidade de transferência de massa gás-líquido e da velocidade de consumo de oxigênio dissolvido. O oxigênio, então, pode ser o fator limitante para o tratamento de águas residuárias utilizando reatores aeróbios.

As etapas de transporte de oxigênio em reatores biológicos com aeração por ar difuso estão representadas na Figura 2. Segundo Bailey e Ollis (1986), a transferência de massa gás-líquido para o oxigênio pode encontrar resistência: no filme gasoso dentro da bolha de ar, próximo a interface gás-líquido; na interface gás-líquido; no filme líquido no meio aquoso, próximo a interface gás-líquido; no meio aquoso; no filme líquido no meio aquoso, próximo a interface líquido-sólido; na interface líquido-sólido, ou seja, na membrana celular dos microrganismos; na fase sólida, ou seja, no citoplasma; e nos sítios de reação bioquímica.

Figura 2. Representação esquemática das etapas de onde ocorre a resistência ao transporte de oxigênio da bolha de ar às células microbianas.

Fonte: adaptado de Bailey e Ollis, 1986.

A resistência de cada etapa dependerá da hidrodinâmica da bolha de gás, da solubilidade do oxigênio, da temperatura, da atividade microbiológica, das substâncias presentes na solução e de fenômenos interfaciais (FAZOLO, 2003). Normalmente, o local que oferece a maior resistência à transferência de massa é a interface gás-líquido, ou seja, este local que determina a limitação ao transporte de oxigênio.

A taxa de transferência de oxigênio para o meio líquido dependerá da resistência à transferência de massa e da solubilidade do oxigênio na água.

Estes processos são influenciados pela interdependência entre diversos fatores, sendo estes: temperatura, concentração de saturação de oxigênio da água, concentração instantânea do oxigênio na água, turbulência, dimensões e geometria do tanque e do sistema de aeração (BAILEY e OLLIS, 1986).

A determinação do coeficiente global de transferência de oxigênio (k_{La}) é fundamental para o dimensionamento de sistemas aeróbios de tratamento de efluentes. O mecanismo de transferência entre as fases líquida e gasosa, proporcionado ou não por sistemas de aeração, pode ser descrito por modelos teóricos ou semi-empíricos.

Destacam-se entre as teorias que descrevem os fenômenos de troca na interface gás-líquido: a teoria dos dois filmes, a teoria da penetração e a teoria da renovação superficial discutidas em Schroeder (1977), Barbosa Júnior (1989) e Shiau (1995).

3.1.1. Teoria dos dois filmes

Esta teoria é aplicável na maioria das situações encontradas, apresentando resultados muito parecidos comparados aos resultados obtidos com teorias mais complexas (FAZOLO, 2003). A teoria dos dois filmes foi apresentada por Lewis e Whitman em 1924.

Considera-se que a interface gás-líquido possui dois filmes, um na fase gasosa e outro na fase líquida. Esses filmes dificultam a transferência de massa, pois considera-se que não existe escoamento nos filmes, ou seja, a transferência ocorre apenas por difusão molecular. Além disso, a parte externa aos filmes é considerada completamente misturada, o que torna os gradientes de concentração desprezíveis.

Como os filmes líquido e gasoso são muito pequenos, considera-se que não há acúmulo de massa neles. Isso implica que o fluxo de massa no filme líquido

($J_{líquido}$) seja igual ao fluxo de massa no filme gasoso (J_{gasoso}). O consumo de oxigênio gera um gradiente de concentração no meio líquido e de pressão na fase gasosa. Assim:

$$J_{gasoso} = k_G \cdot (P_B - P_I) \quad (5)$$

$$J_{líquido} = k_L \cdot (C_I - C_L) \quad (6)$$

em que:

P_B = pressão parcial no meio gasoso [$kg \cdot cm^{-1} \cdot s^{-2}$];

P_I = pressão parcial na interface [$kg \cdot cm^{-1} \cdot s^{-2}$];

C_I = concentração de oxigênio na interface líquida [$mg \cdot L^{-1}$];

C_L = concentração de oxigênio no meio líquido [$mg \cdot L^{-1}$];

k_G e k_L = coeficientes de transferência de massa para os filmes gasoso e líquido [s^{-1}], que correspondem idealmente à D/δ , onde D é a difusividade molecular [$mm \cdot s^{-1}$], e δ é a espessura do filme [mm].

A lei de Henry permite concluirmos que a concentração de gás na interface é proporcional a pressão parcial daquele gás na interface. Assim:

$$P_I = H \cdot C_I \quad (7)$$

$$P_B - P_I = H \cdot (C^* - C_I) \quad (8)$$

$$P_I - P^* = H \cdot (C_I - C_L) \quad (9)$$

em que:

H = constante de Henry;

C^* = concentração de equilíbrio para a pressão parcial do gás P_B ;

P^* = pressão de equilíbrio para a concentração do gás C_L ;

Com as relações obtidas pela lei de Henry, é possível obter uma equação com parâmetros mensuráveis, como a concentração de oxigênio dissolvido na fase líquida. Assim:

$$J = K_L \cdot (C^* - C_L) \quad (10)$$

onde:

$$K_L = \frac{k_g \cdot k_l \cdot H}{k_l + k_g \cdot H} \quad (11)$$

em que:

K_L = coeficiente de transferência de massa global [$mm \cdot s^{-1}$];

k_g = velocidade de transferência de massa no filme gasoso viscoso [$mm \cdot s^{-1}$];

k_l = velocidade de transferência de massa no filme líquido viscoso [$mm \cdot s^{-1}$];

H = constante de Henry [$atm \cdot m^{-3} \cdot mol^{-1}$];

O K_L corresponde à composição dos coeficientes de transferência de massa das etapas do transporte de oxigênio descritas na Figura 2.

Como dificilmente ocorrem escoamentos não-turbulentos capazes de gerar camadas laminares na interface gás-líquido, a teoria dos dois filmes dificilmente é observada. Mas por sua simplicidade, o modelo é utilizado como base para modelos mais avançados.

3.1.2. Teoria da penetração

A teoria da penetração foi postulada inicialmente por Higbie em 1935, que estabeleceu o sistema de dois filmes na interface para um estado transiente, sendo que a etapa controladora do processo de transporte de oxigênio é a passagem pelo filme líquido. Considerando a ausência de reações químicas, tem-se:

$$\frac{\partial C}{\partial t} = D \cdot \frac{\partial^2 C}{\partial z^2} \quad (12)$$

em que:

D = difusividade molecular do oxigênio na água [$mg \cdot mm^{-2}$];

z = distância da interface [mm];

C = concentração de oxigênio dissolvido [$mg \cdot L^{-1}$].

As condições inicial e de contorno para a Equação (12) são:

- $C = C_L$ para $t = 0$;
- $C = C_s$ em $y = 0$, para $t > 0$;
- $C = C_L$ quando $y \rightarrow \infty$.

Considerando uma camada de comprimento infinito e levando em conta apenas as etapas iniciais de transferência de massa, pode-se aplicar as seguintes condições:

- $t = 0$, $z > 0$, $C = C_L$;
- $t > 0$, $z = 0$, $C = C^*$;
- $t > 0$, $z \rightarrow \infty$, $C = C_L$.

Resolvendo a Equação (12) por mudança de variáveis, $C' = C - C_L$, o fluxo na interface pode se calculado como:

$$J = -(C^{**} - C_L) \cdot \sqrt{\frac{D_t}{\pi \cdot t_c}} \quad (13)$$

em que:

C^{**} = concentração instantânea de oxigênio dissolvido [$mg \cdot L^{-1}$];

C_L = concentração de oxigênio dissolvido no líquido [$mg \cdot L^{-1}$];

D_t = difusão turbulenta [$mm^2 \cdot s^{-1}$];

t_c = tempo retenção celular [s].

Analisando a Equação (13), pode-se concluir que a taxa de transferência de massa será maior para tempos de contato mais curtos. Isso pode ser percebido

nos processos de aeração que aumentam a turbulência na interface gás-líquido.

3.1.3. Teoria da renovação superficial

A teoria da renovação superficial, apresentada por Shiau (1995), é uma ampliação da teoria da penetração, pois considerou ainda que porções do líquido estariam na interface em períodos de tempo finitos, distribuídos aleatoriamente devido a turbulência do líquido. Foi proposta a seguinte equação:

$$M = K_L \cdot A \cdot (C_{Si} - C_L) \quad (14)$$

em que:

M = massa de oxigênio transferido por unidade de tempo [$mg \cdot s^{-1}$];

K_L = coeficiente de transferência baseado no filme líquido (relacionado à taxa de renovação superficial) [$mm \cdot s^{-1}$];

A_i = área interfacial para a transferência de massa [mm^2];

C_{Si} = concentração de saturação do oxigênio na interface [$mg \cdot L^{-1}$];

C_L = concentração de oxigênio no líquido [$mg \cdot L^{-1}$].

3.1.4. Fatores que afetam a transferência de oxigênio

A transferência de massa dependerá do sistema de aeração adotado e de condições climáticas, como temperatura e pressão atmosférica. Além disso,

substâncias químicas dispersantes, turbulência e a área destinada à transferência de massa também influenciam. As condições que cada um desses fatores se apresentam no sistema será determinante na obtenção de parâmetros de equacionamento como a concentração de saturação e o coeficiente global de transferência de massa.

3.2. Dificuldades Operacionais em Sistemas de Aeração

Muitos dos problemas operacionais existentes no tratamento de águas residuárias estão relacionados a variações de carga de poluentes afluentes às estações. Na concepção do projeto, o dimensionamento de uma ETE é todo realizado considerando os parâmetros físico-químicos do efluente como constantes ao longo do tempo, porém isso não se torna efetivo durante a operação do sistema.

As ETEs podem sofrer distúrbios externos e internos. Distúrbios externos são resultantes de variações na vazão de entrada, alterações nos parâmetros de qualidade e mudanças de temperatura, por exemplo. Distúrbios internos estão relacionados a má operação do sistema e a erros no dimensionamento hidráulico, por exemplo.

Segundo Costa et al. (2003), o efluente de uma indústria química do município de Belford Roxo, RJ, apresenta grande intervalo de valores para os parâmetros de qualidade. O monitoramento realizado pelos autores mostrou que a DQO (Demanda Química de Oxigênio) variou dentro da faixa de 900 a 2100 mg/L, enquanto a DBO_{5,20} (Demanda Bioquímica de Oxigênio de 5 dias a 20°C) ficou entre 90 e 370 mg/L.

Variações na vazão de entrada das estações de tratamento também prejudicam muito a eficiência do tratamento. A Figura 3 apresenta um histograma de vazões na entrada de um tratamento biológico, mostrando que

as oscilações da vazão de efluente durante 24 horas também são significativas, o que implicará em problemas operacionais no sistema de tratamento de efluentes.

Figura 3. Perfil 24 horas de vazão afluyente a um tanque de aeração recebendo efluente de indústria de embalagens de papel. Fonte: BEAL et al., 2006.

As variações de carga ocasionam impactos significativos no tratamento de efluentes, ora sobrecarregando os reatores, ora com atuação excessiva de aeradores. Em sistemas de aeração, isso implica em gasto excessivo de energia e perda de eficiência no tratamento. Para evitar estas duas situações é necessário atuar no processo de modo a compensar os distúrbios. O controle de processos atua nos parâmetros de processo para evitar sobrecargas e excessos, buscando sempre uma posição ótima de funcionamento do sistema.

Segundo Metcal & Eddy, 2003, quando um sistema de lodos ativados estiver com concentração de oxigênio dissolvido abaixo do valor ótimo pode ocorrer baixa remoção de poluentes, piorar a sedimentabilidade do lodo, turbidez elevada no efluente devido ao floco mal formado e possibilidade de excesso de microrganismos filamentosos. Já em situações de concentrações acima do valor ótimo, haverá aumento nos custos operacionais e quebra do floco devido ao excesso de turbulência ocasionado pelo sistema de aeração.

4. CONTROLE DE PROCESSOS DE AERAÇÃO

Considerar o comportamento dinâmico e incorporar sistemas de controle modernos podem levar a melhorias substanciais no desempenho de plantas de tratamento de águas residuárias, além de trazer benefícios como melhor confiabilidade, redução de custos operacionais e operação estável (ANDREWS, 1974).

Com o controle de processos aplicado ao sistema, espera-se reduzir a variação do parâmetro de processo que será controlado, ou até mesmo torná-lo constante. A Figura 4 ilustra o comportamento de um processo com variação na entrada sem controle e com parâmetros controlados, mostrando que espera-se uma redução na amplitude de oscilação do valor do parâmetro controlado na saída do processo.

Figura 4. Variação no tempo da entrada e da saída de processo (a) não-controlado e (b) controlado. Fonte: adaptado de Andrews, 1974.

4.1. Estratégias de controle

O principal desafio em controle de processos é determinar a alteração necessária na variável manipulada para que a variável controlada seja mantida próxima ao “*set-point*” estabelecido. Para isso existem diversas metodologias, indo das mais elementares, como controle liga-desliga, até metodologias com algoritmos mais sofisticados com predição e adaptação.

As estratégias de controle são fundamentalmente relacionadas com a manipulação de informação, sendo que isto poderá ser feito tanto manualmente quanto por sistemas automáticos de controle.

Em sistemas manuais de controle, é um ser humano que fecha o ciclo da informação, ou seja, uma pessoa faz a leitura da variável controlada na saída do processo, compara ao valor desejado e atua sobre a variável manipulada, abrindo ou fechando uma válvula, por exemplo. É um tipo de controle chamado “*feedback*”, pois a ação de controle é realizada na entrada do sistema após uma leitura na saída do processo.

Quando se trata de um sistema automático de controle, a função do ser humano é substituída por um controlador. O controlador fará a comparação da leitura na saída do processo com o “*set-point*”. Se existir alguma diferença, o controlador calcula, por meio de um algoritmo de controle, a ação sobre a variável manipulada necessária para que a variável controlada alcance o valor desejado. Um sinal é emitido pelo controlador para o elemento final de controle atuar sobre o processo.

Independentemente se o controle é manual ou automático, Andrews (1974) recomenda que para o desenvolvimento de uma estratégia de controle é importante responder algumas questões:

- Quais parâmetros devem ser medidos para iniciação de uma estratégia de controle? Deve-se considerar se a medição será realizada no afluente ou no efluente ao processo, no ambiente ou dentro do processo; as metodologias utilizadas, a frequência e a precisão devem

ser considerados, além da disponibilidade de instrumentação; a dinâmica do sistema também é importante e deve ser levada em consideração.

- Quais as ações de controle poderão ser realizadas? É importante considerar as limitações dos sistemas a serem controlados, principalmente em plantas já em operação, pois normalmente os sistemas são projetados sem considerações adequadas do comportamento dinâmico e das características operacionais; além disso, condições físicas do sistema podem limitar as ações de controle.
- Como a informação será transmitida do sensor para o controlador e do controlador para o elemento final de controle? Deveria ser uma questão simples de ser resolvida, mas muitas vezes a informação coletada nas plantas de tratamento de águas residuárias são apenas para registro o que impede o fechamento da malha de controle.
- Como a informação será processada de modo a determinar o tipo e amplitude da ação de controle necessária? São vários os algoritmos de controle e deve-se estudar qual deles irá atuar melhor sobre cada processo.

Este trabalho tem como foco a última das questões colocadas, e por isso aprofundaremos a discussão apenas sobre os algoritmos de controle. Para os três primeiros questionamentos, poderão ser consultadas obras como Metcalf & Eddy (2003), Seborg et al. (1989) e Stephanopoulos (1983). A partir daqui, quando dissermos controle de processo estaremos nos referindo ao controle automático de processos.

4.2. Estrutura básica para controle de oxigênio dissolvido em sistemas de aeração

A idéia básica de um sistema de controle automático de processos é manter um ou mais parâmetros do processo próximos de um valor estabelecido, chamado “*set-point*”. Para isso é necessário atenuar, ou mesmo anular, o efeito dos distúrbios sobre os parâmetros de interesse. São chamados variáveis controladas aqueles parâmetros que terão seus valores mantidos próximos ao “*set-point*”. Já os parâmetros que são alterados diretamente para que as variáveis controladas alcancem o “*set-point*”, ou seja, anulem os efeitos dos distúrbios, são conhecidos como variáveis manipuladas.

No controle de sistemas de aeração, a principal variável controlada é a concentração de oxigênio dissolvido no tanque de aeração, enquanto a variável manipulada correspondente estará relacionada com o fornecimento de ar/oxigênio. No caso de aeradores por ar difuso, a variável manipulada será a vazão de ar, que pode ser determinada pela abertura de uma válvula ou com a rotação de um soprador. Para aeradores mecânicos superficiais, uma possível variável controlada seria a frequência de rotação das pás.

A estrutura física básica para um sistema de controle de processos de aeração está apresentada na Figura 4. É um sistema malha fechada com controlador por retro-alimentação ou “*feedback*” (METCALF & EDDY, 2003).

Figura 5. Estrutura básica para o controle de oxigênio dissolvido em tanques de aeração de sistemas de tratamento por lodos ativados.
Fonte: adaptado de Metcalf & Eddy, 2003.

Os componentes principais, ou elementos de controles, necessários ao sistema são:

- Aquisição de dados: é composta por sensor de oxigênio dissolvido – para realização de medidas diretas que produzam sinais passíveis de serem transformados em sinais elétricos; transdutor – que converte os sinais emitidos pelo sensor em sinais elétricos, na forma de voltagem ou corrente, ou em sinais pneumáticos; e transmissor – que transforma o sinal emitido pelo transdutor em um sinal padrão para ser enviado a longas distâncias;
- Controlador: é uma central de processamento matemático de dados que converte a variável de saída em variável manipulada sob a lei de controle utilizada;
- Elemento final de controle: recebe o sinal do controlador e atua sobre os aeradores de forma a proporcionar o movimento adequado da variável manipulada.

A aquisição de dados de concentração de oxigênio dissolvido (OD) é realizada por um sensor, acoplado a um transdutor que retorna medidas em miligramas por litro ($mg \cdot L^{-1}$) ou partes por milhão (ppm). As leituras são encaminhadas para o controlador que compara o valor medido de OD com o “set-point” estabelecido e calcula a próxima ação de controle. Estabelecida a ação de

controle, o controlador emite um sinal para o elemento final de controle que irá atuar sobre o processo, sendo que este elemento poderá ser uma válvula atuadora ou um inversor de frequência, ambos com objetivo de controlar a vazão de ar do soprador.

Para obter uma resposta em circuito fechado adequada do processo de aeração em operação com sistema de controle, o controlador deve ser sintonizado a um dado mecanismo de ação. Desta forma, Seborg et al. (1989), citado por Rodrigo et al. (1999) e por Zhang et al. (2006), estabelecem seis critérios de performance que um sistema de circuito fechado deve satisfazer:

- o sistema de circuito fechado deve ser estável;
- os efeitos de perturbação devem ser minimizados;
- respostas rápidas e estáveis para mudanças no setpoint devem ser obtidas;
- “*off-sets*” ou desvios do “*set-point*” devem ser eliminados;
- ações de controle excessivo devem ser evitadas; e
- o sistema de controle deve ser robusto, ou seja, suportar adequadamente as mudanças nas condições de processo e a erros no modelo de processo adotado.

Os movimentos de controle para atenuação dos distúrbios devem ser bastante precisos, pois mudanças inadequadas podem prejudicar o processo. Vários são os algoritmos de controle para o cálculo dos movimentos da variável manipulada.

5. ALGORITMOS DE CONTROLE

Controladores “*feedback*” podem atuar com algoritmos clássicos PID (Proporcional, Integrativo, Derivativo), como também com algoritmos que utilizam modelos do processo. No controle clássico PID, o cálculo das ações de controle é baseado na diferença (ou erro) entre a leitura e o valor desejado da variável controlada. Já o controle baseado em modelo (MBC, *Model Based Control*) calcula as ações de controle utilizando diretamente um modelo do processo.

Corrêa (2006) mostra que o controle PID é satisfatório quando utilizado para o controle de processo de aeração em corpos d’água, porém ressalta a necessidade de utilização de outros algoritmos para melhoria do controle do processo. Além disso, Corrêa et al. (2006) mostraram que o controle PI em processos de secagem (que fazem analogia aos processos de aeração) não é interessante por acarretar grandes diferenças na variável controlada, representadas por grandes sobre-elevações, indicando que estudar o MBC em processos de aeração pode ser uma alternativa interessante.

No MBC, uma das técnicas consagradas na indústria de processos é o controle preditivo com modelo (MPC, *Model Predictive Control*), com robustez satisfatória para controle com múltiplas entradas e múltiplas saídas (MIMO, *Multiple Input – Multiple Output*), aceitando inclusive restrições nas variáveis manipuladas e nas variáveis controladas.

A característica fundamental do MPC é que o algoritmo de controle prevê o comportamento futuro da variável controlada por meio de um modelo dinâmico do processo e das medidas da variável controlada (Figura 6).

Figura 6. Ações passadas, atual e o futuro predito do processo controlado.
Fonte: adaptado de García et al. (1989).

A partir da minimização de uma função objetivo, a sequência de movimentos de controle (L , horizonte de controle) é calculada. Em cada instante amostral (k), os cálculos são revistos e as previsões são atualizadas em um horizonte definido (R , horizonte de otimização) por meio de um modelo de convolução discreto do processo. É importante utilizar o horizonte móvel de controle pela dificuldade em garantir que os movimentos de controle preditos no instante “ k ” estarão corretos no momento de sua aplicação. Isto devido aos erros de modelagem e aos distúrbios não controlados do processo.

Em estudo realizado por Corrêa et al. (2006), comparando diversos algoritmos de controle, os autores concluíram que os algoritmos do tipo DMC (*Dynamic Matrix Control*) e QDMC (*Quadratic Dynamic Matrix Control*) possuem características específicas muito interessantes para aplicação em controle de processos de aeração.

O DMC é um algoritmo de controle multivariável sem restrições desenvolvido por Cutler e Ramaker (1980). Segundo Qin e Badgwell (2003), aspectos importantes desse algoritmo são:

- Utiliza modelo linear de resposta ao degrau;
- O objetivo da performance quadrática é sobrepor um horizonte de predição finito;
- O comportamento futuro da variável controlada é descrito seguindo o valor desejado o mais próximo possível;

- A solução ótima do problema de mínimos quadrados é computada como entrada da planta.

É interessante para o sistema restringir os valores da variável manipulada dentro de uma faixa de trabalho aceitável dando mais estabilidade para o processo. Essas restrições podem ser definidas como:

$$\mathbf{u}_{\min} \leq \mathbf{u} \leq \mathbf{u}_{\max} \quad (15)$$

$$\Delta \mathbf{u}_{\min} \leq \Delta \mathbf{u} \leq \Delta \mathbf{u}_{\max} \quad (16)$$

em que:

- $\mathbf{u}_{\min}$ = limite mínimo para a variável manipulada;
- $\mathbf{u}_{\max}$ = limite máximo para a variável manipulada;
- $\mathbf{u}$ = variável manipulada;
- $\Delta \mathbf{u}_{\min}$ = movimento mínimo de controle;
- $\Delta \mathbf{u}_{\max}$ = movimento máximo de controle;
- $\Delta \mathbf{u}$ = movimento de controle;

Para incorporar as restrições na variável manipulada é preciso utilizar uma técnica de otimização como a programação quadrática (QP do inglês “*Quadratic Programming*”).

O QDMC foi descrito por García e Morshedi (1986), que atribuíram uma programação quadrática ao DMC deixando explícitas as restrições na variável manipulada. Segundo Qin e Badgwell (2003), aspectos importantes desse algoritmo são:

- Utiliza modelo linear de resposta ao degrau;
- O objetivo da performance quadrática é sobrepor um horizonte de predição finito;
- O comportamento futuro da variável controlada é descrito seguindo o valor desejado o mais próximo possível submetido a um termo de supressão de movimento;
- A solução ótima da programação quadrática é computada como entrada da planta.

6. APLICAÇÕES

Neste item serão apresentados trabalhos não só de controle de aeração, mas também de aplicação de controladores em condições semelhantes aquelas estabelecidas para este trabalho, visto, inclusive a deficiência de trabalhos semelhantes ao aqui apresentado.

Lee *et al.* (1998) utilizaram controlador proporcional-integral (PI) em sistema de controle automático de oxigênio dissolvido e pH em lodos ativados de indústrias de coque. O sistema foi testado inicialmente em uma planta piloto e houve redução de 70% do consumo de energia dos aeradores superficiais, que mantinham 2 mg/L de oxigênio dissolvido no tanque de aeração. No teste em escala real, o controlador mostrou bom desempenho, mantendo boa aderência ao valor de 2 mg/L no aerador. Além disso, houve redução de 40% no consumo de energia elétrica pelos aeradores e melhoria significativa na qualidade do efluente tratado.

Testes em escala real realizados por Corrêa *et al.* (2004), aplicando controladores QDMC em leitos de secagem, mostraram que este controlador é bastante robusto para manter os valores estáveis das variáveis controladas e é eficiente em limitar a operação do sistema dentro dos limites dos equipamentos. Além disso, o controlador se mostrou eficaz independentemente da escala do processo a ser controlado nos casos estudados.

Aplicando controle de aeração otimizado em uma planta de lodos ativados com aeração alternada, Chachuat *et al.* (2005) mostraram que, mesmo com controle em malha aberta, os ganhos de eficiência de remoção de amônia e de economia de energia são bastante significativos. Os problemas de otimização considerados pelos autores foram a minimização da descarga de nitrogênio no corpo receptor com ganho de confiabilidade da planta e a redução de custos operacionais da planta. Os autores concluíram que reduziram em 37% a descarga de nitrogênio do ambiente e reduziram em 27% o gasto com energia

da planta. Eles ainda concluem que um sistema em malha fechada pode alcançar resultados ainda melhores.

Em estudo de simulação, Caraman *et al.* (2007) aplicaram controle preditivo utilizando uma rede neural como modelo interno para controlar a concentração de substrato orgânico em um sistema de lodos ativados convencional. Os autores mostraram que o controle preditivo baseado em modelo de rede neural é robusto para o controle de oxigênio dissolvido, mesmo em condições diversas de operação da planta, visto em todas as simulações a concentração final de substrato no efluente estava enquadrada nos limites de legislação.

Em outro estudo de simulação, Holanda *et al.* (2008) testaram dois exemplos de aplicação para lodos ativados: um primeiro sistema com nitrificação e pré-desnitrificação, para remoção de nitrogênio, e outro sistema com aeração alternada, também visando a remoção de nitrogênio. Pelos resultados da simulação, os autores concluíram que o controle preditivo por modelo pode ser efetivamente aplicado no controle de oxigênio dissolvido em plantas de lodos ativados.

Utilizando dados reais de um sistema de lodos ativados com remoção de nitrogênio e fósforo, Chotkowski *et al.* (2005) realizaram estudo de simulação para testar dois controladores: um controlador preditivo por modelo não-linear e um controlador adaptativo por referência direta. O controlador adaptativo se mostrou mais simples para desenvolvimento, porém o controlador preditivo mostrou melhor desempenho e robustez.

Öskan *et al.* (2008) aplicaram controladores por matriz dinâmica em um sistema de controle de pH com processo de neutralização com ácido sulfúrico e carbonato de cálcio. Foram testados dois controladores: um com modelo linear, de resposta ao degrau, e outro com modelo não-linear, tipo NARMAX (sigla para “*Non-Linear Auto-Regressive Moving Average with Exogeneous Input*”). Os autores concluíram que o controlador com modelo não-linear tem melhores resultados que o controlador com modelo linear, porém, é fundamental que o modelo não-linear seja determinado corretamente. Assim, o controlador com modelo linear tem a vantagem de ser de mais simples implementação.

7. MATERIAL E MÉTODOS

7.1. Aparato Experimental

A planta experimental (Figura 7) está instalada no Laboratório de Hidráulica Ambiental do Centro de Recursos Hídricos e Ecologia Aplicada (CRHEA), Escola de Engenharia de São Carlos (EESC/USP), localizado nas proximidades da represa do Broa, no município de Itirapina, SP. É composta de um aerador por ar difuso de fluxos cruzados sem agitação mecânica. Trata-se de um canal aberto de 5 m de comprimento, 350 mm de altura e 200 mm de largura, em material acrílico, para visualização das plumas de bolhas de ar. Para alimentação do tanque durante os ensaios foi usada água da rede de abastecimento.

Figura 7. Planta experimental do tanque de aeração, dimensões e disposição do sistema para aquisição de dados e controle.

A vazão de entrada de água no canal é medida por um rotâmetro (Figura 8). Os aeradores estão fixados na linha central, no fundo, nos 3 m finais do canal, espaçados igualmente entre si (10 cm) e dispostos longitudinalmente. A água sai do canal por transbordamento em um bocal apropriado.

Figura 8. Fotografia do aparato experimental instalado no laboratório, com destaque para a bomba de recirculação de água, o rotâmetro e a dosagem de hipossulfito de sódio.

A planta experimental possui um sensor de oxigênio dissolvido instalado na saída do canal de aeração. Este sensor está ligado a um transmissor e a um condicionador de sinais para adequação do sinal para entrada na interface AD, acoplada ao microcomputador. O sinal é enviado ao controlador, que processa essa informação e emite um comando para um inversor de frequência que regula a rotação do soprador de ar para controlar a vazão de ar do sistema de aeração.

Para economia de água, foi utilizado um sistema de recirculação usando uma bomba centrífuga de 0,75 cv. Para redução da concentração de oxigênio na água de recirculação era dosado sulfito de sódio na entrada do canal de recirculação, que reduzia para aproximadamente 1 ppm a concentração de oxigênio dissolvido. O consumo de oxigênio ocorria segundo a equação:

onde:

$S_3O_4^{2-}$ = representação da molécula do íon hipossulfito;

O_2 = representação da molécula de oxigênio;

SO_4^{2-} = representação da molécula do íon sulfato.

O experimento foi realizado sempre em temperatura ambiente. Não houve qualquer tipo de controle em relação a este parâmetro. A temperatura medida na água esteve sempre entre 24 e 25°C, e por isso considerou que este parâmetro não influenciou no processo e portanto não haverá análise a respeito de seu comportamento e sua influência na dinâmica do processo.

7.2. Determinação da faixa operacional do soprador

O controlador proposto permite estabelecer uma faixa de operação para os equipamentos, ou seja, possibilita restringir as variáveis manipuladas em valores máximos e mínimos, respeitando os limites físicos dos atuadores ou de qualidade operacional (evitando instabilidades, “*off-sets*”, entre outros).

Neste trabalho, a faixa operacional foi determinada pensando na mínima rotação do soprador na qual os bicos de aeração não se afogariam e na máxima rotação em que o soprador era capaz de alcançar.

Para determinar a máxima rotação, o inversor de frequência foi estimulado até o ponto a partir do qual a rotação não mais se alterou, sendo fixado então o limite superior de rotação, que foi de 3500 rpm. Já para a mínima rotação, o inversor de frequência foi estimulado até a situação de equilíbrio quando não ocorreu mais o borbulhamento de ar, ou seja, a vazão de ar estabeleceu um

equilíbrio com a pressão hidrostática do canal, sendo que a rotação determinada foi de 900 rpm.

Estabelecidos os limites máximo e mínimo de rotação do soprador de ar, foi possível inserir no controlador dados que limitaram sua atuação, evitando atuação excessiva do sistema.

7.3. Identificação do modelo do processo de aeração

O modelo de processo adotado foi o modelo de primeira ordem em resposta ao degrau na variável controlada. Para a identificação do modelo, foram realizados ensaios com perturbação degrau na rotação do soprador. O modelo é dado pela seguinte equação:

$$Y = Y_i + K \cdot A \cdot \left[1 - e^{\left(\frac{\theta - t}{\tau} \right)} \right] \quad (18)$$

onde:

Y = resposta ao estímulo [ppm];

Y_i = estado estacionário inicial [ppm];

K = ganho do processo [$\frac{ppm}{rpm}$];

A = amplitude do degrau na variável de entrada [rpm];

t = tempo [s];

θ = tempo de atraso [s];

τ = constante de tempo [s].

Para a determinação dos parâmetros do modelo foram realizados teste em malha aberta com perturbação degrau na rotação do soprador. A Tabela 2 mostra a amplitude do degrau e o número de experimentos para cada amplitude testada.

Tabela 1. Características dos experimentos realizados para identificação do modelo de processo.

Degrau (rpm)	Rotação inicial (rpm)	Rotação final (rpm)
- 300	1400	1100
- 300	1700	1400
- 300	2000	1700
- 300	2600	2300
- 750	2250	1500
+ 300	1400	1700
+ 300	1700	2000
+ 600	2000	2600
+ 750	2250	3000

A partir dos dados obtidos nos ensaios em malha aberta realizados, foi possível identificar o modelo de resposta ao degrau correspondente ao processo de aeração estabelecido no aparato experimental em questão.

Cada experimento teve duração aproximada de 1000 segundos, de forma a alcançar um novo estado estacionário. Durante a realização dos experimentos, percebeu-se que este tempo era excessivo e por isso trabalhou-se apenas com os dados de aproximadamente 400 segundos.

O tempo morto em cada operação realizada em malha aberta foi determinado visualmente após análise dos dados em gráficos. O estado estacionário inicial foi determinado por meio da média das leituras realizadas até o final do tempo morto.

Com os dados dispostos em gráficos, realizou-se uma regressão pelo modelo de primeira ordem em resposta a perturbação degrau e determinou-se os parâmetros do modelo empírico do processo. Para cada experimento em malha aberta foi realizada uma regressão desse tipo. O valor de cada parâmetro foi obtido segundo a média dos valores encontrados em casa regressão.

7.4. Controlador

A implementação do controle foi realizada em sistema SISO (*“simple input, simple output”*). A variável controlada foi a concentração de oxigênio dissolvido (C) na saída do canal de aeração. A variável manipulada foi a rotação do soprador (rot), a fim de controlar vazão de ar (Q_{ar}). A vazão de água na entrada do canal (Q) foi considerada como variável distúrbio, pois em sistemas de tratamento de efluentes este é um dos principais parâmetros causadores de distúrbios.

A implementação da técnica de controle seguiu a estrutura do controle MPC (*“model predictive control”* ou controle preditivo com modelo). No caso do controle preditivo DMC (Cutler e Ramaker, 1979), uma matriz dinâmica será construída a partir da resposta frente ao estímulo degrau na variável manipulada. Considerou-se o problema de restrição na variável manipulada rotação do soprador (rot). Para resolver isso, foi aplicado um método de programação quadrática no algoritmo DMC, tornando algoritmo QDMC (*“quadratic dynamic matrix control”*).

É importante ressaltar que os programas de monitoração e controle “on-line” foram elaborados com rotinas em linguagem C para implementação em tempo real. Estas rotinas foram adaptadas dos trabalhos desenvolvidos por Corrêa (2000).

A Figura 9 mostra o diagrama de blocos para o funcionamento do controlador em estudo.

Figura 9. Funcionamento do controlador aplicado ao canal de aeração em estudo.

7.5. Descrição matemática do algoritmo de controle

A formulação do algoritmo DMC parte de um modelo de convolução discreta dado por:

$$\hat{y}_{k+1} = \hat{y}_k + \sum_{i=1}^N h_i \cdot \Delta u_{k+1-i} \quad (19)$$

onde:

$\hat{y}_{k+1}$ = valor predito para a variável controlada no tempo $k + 1$;

$\hat{y}_k$ = valor predito para a variável controlada no tempo k ;

h_i = valor do pulso no i -ésimo intervalo de tempo discreto;

Δu_{k+1-i} = ação controladora no tempo $k + 1 - i$;

N = número de termos da seqüência de resposta ao degrau ou horizonte do modelo.

O valor do pulso foi obtido dando um degrau na variável manipulada. A Figura 10 mostra como foram obtidos os valores do pulso, possibilitando a construção da matriz dinâmica do processo.

Figura 10. Resposta ao degrau na variável manipulada.

O controlador DMC busca minimizar o erro dentro do horizonte de predição (R), o qual será menor que o horizonte do modelo ($R < N$).

Os valores da variável controlada obtido no processo (y_k) realimentam o controlador para que a predição seja corrigida recursivamente, gerando um valor corrigido para a variável controlada. Isso altera o modelo de convolução para:

$$y_{k+j}^c = y_{k+j-1} + \sum_{i=1}^N h_i \cdot \Delta u_{k+j-1} \quad (20)$$

onde:

$$j = 1, 2, \dots, R$$

y_{k+j}^c = predição corrigida da variável controlada no tempo $k + j$;

y_{k+j-1} = valor medido da variável controlada no tempo $k + j - 1$;

R = horizonte de predição.

Fazendo $y_k^c = y_k$ e substituindo cada equação (y_k^c) na equação do instante posterior (y_{k+1}^c), tem-se:

$$\left\{ \begin{array}{l} y_{k+1}^c = a_1 \cdot \Delta u_k + [y_k + P_1] \\ y_{k+2}^c = a_2 \cdot \Delta u_k + a_1 \cdot \Delta u_{k+1} + [y_k + P_2] \\ \vdots \\ y_{k+R}^c = a_R \cdot \Delta u_k + a_{R-1} \cdot \Delta u_{k+1} + a_{R-2} \cdot \Delta u_{k+2} + \dots + a_1 \cdot \Delta u_{k+R-1} + [y_k + P_R] \end{array} \right. \quad (21)$$

onde:

$$\begin{aligned} a_i &= \sum_{j=1}^i h_j ; \\ P_i &= \sum_{m=1}^i S_m ; \\ S_m &= \sum_{i=m+1}^N h_i \cdot \Delta u_{k+m-1} ; \end{aligned}$$

Passando a Equação 5 para forma matricial, tem-se:

$$\begin{bmatrix} y_{k+1}^c \\ y_{k+2}^c \\ \vdots \\ y_{k+R}^c \end{bmatrix}_{R \times 1} = \begin{bmatrix} a_1 & 0 & \dots & 0 \\ a_2 & a_1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_R & a_{R-1} & \dots & a_1 \end{bmatrix}_{R \times R} \cdot \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+R-1} \end{bmatrix}_{R \times 1} + \begin{bmatrix} y_k + P_1 \\ y_k + P_2 \\ \vdots \\ y_k + P_R \end{bmatrix}_{R \times 1} \quad (22)$$

Para saber qual será a ação de controle a ser implementada, é necessário conhecer o valor desejado para a variável controlada no próximo instante de amostragem, ou seja, saber qual será a trajetória de referência. Isto pode ocorrer selecionando um “*set-point*” ou utilizando o filtro de valores.

$$y_{k+j}^d = \alpha^j \cdot y_k + (1 - \alpha^j) \cdot y_k^{sp} \text{ com } j = 1, 2, \dots, R \text{ e } 0 \leq \alpha < 1 \quad (23)$$

onde:

$$\begin{aligned} y_{k+j}^d &= \text{valor desejado para a variável controlada no próximo instante de tempo;} \\ \alpha^j &= \text{velocidade de aproximação do “set-point”;} \\ y_k &= \text{valor medido para a variável controlada no tempo } k ; \\ y_k^{sp} &= \text{“set-point” da variável manipulada.} \end{aligned}$$

Fazendo a diferença entre a Equação 23 e a Equação 22, obtém-se:

$$\begin{bmatrix} y_{k+1}^d - y_{k+1}^c \\ y_{k+2}^d - y_{k+2}^c \\ \vdots \\ y_{k+R}^d - y_{k+R}^c \end{bmatrix}_{R \times 1} = - \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ a_2 & a_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_R & a_{R-1} & \cdots & a_1 \end{bmatrix}_{R \times R} \cdot \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+R-1} \end{bmatrix}_{R \times 1} + \begin{bmatrix} (1-\alpha^1)E_k - P_1 \\ (1-\alpha^2)E_k - P_2 \\ \vdots \\ (1-\alpha^R)E_k - P_R \end{bmatrix}_{R \times 1} \quad (24)$$

onde:

$$E_k = (y_k^{sp} - y_k)$$

Como o DMC irá determinar o horizonte de controle (L), ele obterá os L movimentos futuros do controlador (Δu), sendo que o horizonte de controle é menor que o horizonte de predição ($L < R$). Essa estratégia faz que o vetor Δu mude da dimensão $R \times 1$ para a dimensão $L \times 1$, ou seja, haverá mais equações do que incógnitas.

A Equação 24 fica então como segue:

$$\begin{bmatrix} y_{k+1}^d - y_{k+1}^c \\ y_{k+2}^d - y_{k+2}^c \\ \vdots \\ y_{k+R}^d - y_{k+R}^c \end{bmatrix}_{R \times 1} = - \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ a_2 & a_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_R & a_{R-1} & \cdots & a_{R-L+1} \end{bmatrix}_{R \times L} \cdot \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+L-1} \end{bmatrix}_{L \times 1} + \begin{bmatrix} (1-\alpha^1)E_k - P_1 \\ (1-\alpha^2)E_k - P_2 \\ \vdots \\ (1-\alpha^R)E_k - P_R \end{bmatrix}_{R \times 1} \quad (25)$$

Pode-se reescrever a Equação 25, tendo:

$$\hat{\mathbf{E}} = -\mathbf{A} \cdot \Delta \mathbf{u} + \hat{\mathbf{E}}' \quad (26)$$

onde:

$\mathbf{A}$ = matriz dinâmica com dimensão $R \times L$;

$\hat{\mathbf{E}}'$ = vetor que representa o desvio entre o “set-point” e a predição da variável controlada;

O vetor $\hat{\mathbf{E}}'$ é calculado a partir da ação de controle passada e representa o desvio entre “set-point” e a predição da variável controlada.

Na Equação 26 são obtidas R equações para encontrar L incógnitas. Como $L < R$, não é possível encontrar uma solução exata para o sistema. Para encontrar a melhor solução busca-se a minimização do índice de performance (função objetivo, $\mathbf{J}(\Delta \mathbf{u})$) pelo método dos mínimos quadrados, sendo:

$$J(\Delta \mathbf{u}) = \hat{\mathbf{E}}^T \cdot \hat{\mathbf{E}} = (\hat{\mathbf{E}}' - \mathbf{A} \cdot \Delta \mathbf{u})^T \cdot (\hat{\mathbf{E}}' - \mathbf{A} \cdot \Delta \mathbf{u}) \quad (27)$$

A solução para a Equação 27 é dada por:

$$\Delta \mathbf{u} = (\mathbf{A}^T \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^T \cdot \hat{\mathbf{E}}' \quad (28)$$

onde:

$(\mathbf{A}^T \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^T$: matriz pseudo-inversa (PENROSE, 1956; GREVILLE, 1977, citados por Marchetti et al., 1983).

Quando a matriz pseudo-inversa for mal condicionada, as ações de controle previstas podem resultar em movimentos bruscos da variável manipulada ($\Delta \mathbf{u}$). Uma alternativa é modificar o índice de performance inserindo uma matriz que penalize movimentos da variável manipulada (CUTLER e RAMAKER, 1980, citados por Marchetti et al., 1983). Tem-se então uma nova função objetivo, dada por:

$$J(\Delta \mathbf{u}) = \hat{\mathbf{E}}^T \cdot \mathbf{Q} \cdot \hat{\mathbf{E}} + \Delta \mathbf{u}^T \cdot \mathbf{R} \cdot \Delta \mathbf{u} \quad (29)$$

onde:

$\mathbf{Q}$ e $\mathbf{R}$: matrizes de ponderação definida e positiva.

Conseqüentemente, tem-se uma nova solução:

$$\Delta \mathbf{u} = (\mathbf{A}^T \cdot \mathbf{Q} \cdot \mathbf{A} + \mathbf{R})^{-1} \cdot \mathbf{A}^T \cdot \mathbf{Q} \cdot \hat{\mathbf{E}}' \quad (30)$$

A Equação 30 fornece as ações controladores em L instantes futuros para cada instante do presente, porém o normal é implementar apenas o primeiro movimento da variável manipulada previsto, sendo que as previsões de ações controladores são recalculadas sempre que ocorre uma amostragem na saída do processo.

Uma opção mais robusta e interessante para restringir os movimentos da variável manipulada é utilizar o QDMC. Ele utiliza as mesmas equações do DMC (Equação 26), porém na função objetivo aplica-se programação

quadrática para chegar a uma solução. A função objetivo então muda para o seguinte equacionamento:

$$J(\Delta \mathbf{u}) = \frac{1}{2} \cdot \hat{\mathbf{E}}^T \cdot \mathbf{Q}^T \cdot \mathbf{Q} \cdot \hat{\mathbf{E}} + \frac{1}{2} \Delta \mathbf{u}^T \cdot \mathbf{R} \cdot \Delta \mathbf{u} \quad (31)$$

A Equação 31 pode ser reescrita como:

$$J = \frac{1}{2} \cdot (y_{sp} - y)^T \cdot \mathbf{Q}^T \cdot \mathbf{Q} \cdot (y_{sp} - y) + \frac{1}{2} \cdot \Delta \mathbf{u}^T \cdot \mathbf{R} \cdot \Delta \mathbf{u} \quad (32)$$

O problema inicial de minimização é representado então por:

$$\min_{\Delta \mathbf{u}} J = (y_{sp} - y)^T \cdot \mathbf{Q}^T \cdot \mathbf{Q} \cdot (y_{sp} - y) + \Delta \mathbf{u}^T \cdot \mathbf{R} \cdot \Delta \mathbf{u} \quad (33)$$

Sujeito às seguintes restrições nas variáveis manipuladas:

$$\begin{aligned} \Delta \mathbf{u}_{\min} &\leq \Delta \mathbf{u} \leq \Delta \mathbf{u}_{\max} \\ \mathbf{u}_{\min} &\leq \mathbf{u} \leq \mathbf{u}_{\max} \end{aligned} \quad (34)$$

Sendo que a variável controlada y é dada pelo modelo:

$$y = \mathbf{G} \cdot \Delta \mathbf{u} + \bar{y} \quad (35)$$

onde:

$\mathbf{G}$ = matriz dinâmica;
 $\bar{y}$ = vetor de respostas passadas.

Para encaixar o problema em um algoritmo de QP, a função objetivo deve ser alterada na estrutura (continuando a mesma na essência) de modo que somente variáveis não negativas sejam introduzidas nos cálculos. Logo, a solução também será somente não-negativa. Introduz-se, então uma variável de trabalho $x \geq 0$ obtida da seguinte transformação:

$$x = \Delta \mathbf{u}_{\max} - \Delta \mathbf{u} \quad (36)$$

ou

$$\Delta \mathbf{u} = \Delta \mathbf{u}_{\max} - x \quad (37)$$

Assim, o modelo pode ser reescrito como:

$$y = \mathbf{G} \cdot \Delta \mathbf{u}_{\max} - \mathbf{G} \cdot x + \bar{y} \quad (38)$$

A função objetivo, após a modificação com essas variáveis, fica:

$$J = \left(y_{sp} - \mathbf{G} \cdot \Delta \mathbf{u}_{\max} + \mathbf{G} \cdot x - \bar{y} \right)^T \cdot \mathbf{Q}^T \cdot \mathbf{Q} \cdot \left(y_{sp} - \mathbf{G} \cdot \Delta \mathbf{u}_{\max} + \mathbf{G} \cdot x - \bar{y} \right) + \left(\Delta \mathbf{u}_{\max} - x \right)^T \cdot \mathbf{R} \cdot \left(\Delta \mathbf{u}_{\max} - x \right) \quad (39)$$

Após rearranjo das variáveis e simplificações, ela tem a seguinte forma:

$$J = \left[\left(y_{sp} - \bar{y} \right)^T \cdot \mathbf{Q} \cdot \mathbf{G} + \Delta \mathbf{u}_{\max}^T \cdot \mathbf{R} \cdot x \cdot \left(\mathbf{G}^T \cdot \mathbf{Q} \cdot \mathbf{G} + \mathbf{R} \right) \right] \cdot x - \frac{1}{2} \cdot x^T \cdot \left(\mathbf{G}^T \cdot \mathbf{Q} \cdot \mathbf{G} + \mathbf{R} \right) \cdot x \quad (40)$$

Em uma forma mais simplificada:

$$J = \mathbf{a}^T \cdot x - \frac{1}{2} \cdot x^T \cdot \mathbf{b} \cdot x \quad (41)$$

Quanto às restrições, elas também devem ser rearranjadas com a variável x , ou seja substituindo (37) em (34), tem-se:

$$\begin{aligned} x &\geq 0 \\ x &\leq \Delta \mathbf{u}_{\max} - \Delta \mathbf{u}_{\min} = r_1 \\ \mathbf{N} \cdot x &\leq \mathbf{N} \cdot \Delta \mathbf{u}_{\max} - \mathbf{u}_{\min} = r_2 \\ -\mathbf{N} \cdot x &\leq -\mathbf{N} \cdot \Delta \mathbf{u}_{\max} + \mathbf{u}_{\max} = r_3 \end{aligned} \quad (42)$$

onde:

$\mathbf{N}$ = matriz $L \times L$ (L – horizonte de controle) triangular inferior unitária.

Na forma matricial, tem-se:

$$\begin{bmatrix} \mathbf{I} \\ \mathbf{N} \\ -\mathbf{N} \end{bmatrix} \cdot x \leq \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} \quad (43)$$

A Equação (43) é equivalente à:

$$\mathbf{A} \cdot x \leq r \quad (44)$$

Logo, o problema de otimização é colocado como:

$$\min_x J = \mathbf{a}^T \cdot x - \frac{1}{2} \cdot x^T \cdot \mathbf{b} \cdot x \quad (45)$$

onde:

$$\mathbf{a}^T = (y_{sp} - \bar{y})^T \cdot \mathbf{Q} \cdot \mathbf{G} - \mathbf{u}_{\min}^T \cdot \mathbf{b} \quad (46)$$

$$\mathbf{b} = \mathbf{G}^T \cdot \mathbf{Q} \cdot \mathbf{G} + \mathbf{R} \quad (47)$$

A Equação (45) está sujeita a Equação (44) tal que $x \geq 0$.

Note que $\mathbf{b}$ deve ser calculado apenas uma vez antes de iniciar o controle. Já $\mathbf{a}$ e r devem ser calculados “on-line” a cada intervalo de amostragem.

Faz-se necessário tornar a restrição da Equação (44) em uma restrição de igualdade com a introdução da “variável de folga” (relaxamento ou tolerância) não-negativa v , ou:

$$\mathbf{A} \cdot x + v = r \quad (48)$$

O valor ótimo de x é obtido pela solução de:

$$\begin{bmatrix} -\mathbf{b} & 0 & -\mathbf{I} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{I} & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ v \\ \mu \\ v \end{bmatrix} = \begin{bmatrix} -\mathbf{a} \\ r \end{bmatrix} \quad (49)$$

onde:

μ e ν = vetores de multiplicadores de Lagrange.

Dessa forma este sistema está pronto para ser inserido em algum algoritmo de Programação Quadrática. Um opção é transformar o problema em Programação Linear usando o método Simplex, conforme Sposito, 1975. A formulação do QDMC, bem como testes, aplicações, estudos de robustez, comparação de performance e controle com restrição podem ser encontrada em Ricker (1985), Garcia e Morshedi (1986), Zafiriou e Marchal (1991), Almeida Neto et al. (2000).

8. RESULTADOS E DISCUSSÃO

O controlador foi desenvolvido em linguagem C para experimentação em escala piloto. No Anexo I encontra-se a linguagem de programação que gerou o programa utilizado como controlador do processo em estudo. A Figura 11 mostra uma fotografia da tela quando o programa estava em operação.

Figura 11. Interface gráfica da programação utilizada no experimento.

O canal de aeração durante a operação do sistema é ilustrado nas Figuras 12 e 13.

Figura 12. Visão lateral do canal de aeração em operação.

Figura 13. Visão da parte superficial da água no canal de aeração em operação.

Nota-se, nas Figuras 12 e 13, que as bolhas de ar chegaram a superfície com tamanho consideravelmente grande e que sua distribuição não foi uniforme. Além disso, percebeu-se a grande turbulência no canal na região em que ocorre a aeração.

8.1. Identificando o modelo do processo de aeração

Os testes em malha aberta com perturbação do tipo degrau na variável manipulada foram realizados na planta experimental sendo obtidos os gráficos apresentados nas Figuras 14 a 24.

Figura 14. Resultado da perturbação degrau negativo no “set-point” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.

Figura 15. Resultado da perturbação degrau negativo no “set-point” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.

Figura 16. Resultado da perturbação degrau negativo no “set-point” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.

Figura 17. Resultado da perturbação degrau negativo no “set-point” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.

Figura 18. Resultado da perturbação degrau negativo no “set-point” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.

Figura 19. Resultado da perturbação degrau negativo no “set-point” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.

Figura 20. Resultado da perturbação degrau negativo no “set-point” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.

Figura 21. Resultado da perturbação degrau positivo no “set-point” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.

Figura 22. Resultado da perturbação degrau positivo no “set-point” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.

Figura 23. Resultado da perturbação degrau positivo no “set-point” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.

Figura 24. Resultado da perturbação degrau positivo no “set-point” realizado para determinação dos parâmetros de entrada do modelo de processo estudado.

Com os dados obtidos nos testes em malha aberta com perturbação degrau na rotação do soprador, foi possível estimar os valores dos parâmetros de entrada no modelo de primeira ordem, assumindo que o processo de aeração em questão possui este comportamento.

É notável nas Figuras 16, 17 e 18 que o processo possui bastante instabilidade, dificultando o processo de modelagem, principalmente quando se trata da perturbação negativa na rotação do soprador. Estes resultados não foram descartados na determinação do modelo de primeira ordem, visto a quantidade de experimentos realizados na identificação do modelo.

A Tabela 3 mostra os dados obtidos nos testes com degrau negativo e com degrau positivo no “*set-point*”, utilizando os valores médios como estimativa para os parâmetros do modelo de processo.

Tabela 2. Resultados dos testes de perturbação degrau no set-point para identificação do modelo de processo.

Degrau (rpm)	Rotação inicial (rpm)	Rotação final (rpm)	OD _{inicial} (ppm)	K (ppm/rpm)	θ (s)	τ (s)
- 300	1400	1100	6,71	$1,67 \times 10^{-4}$	10	125,68
- 300	1700	1400	6,83	$1,67 \times 10^{-4}$	10	104,70
- 300	1700	1400	6,17	$4,33 \times 10^{-4}$	10	86,23
- 300	1700	1400	6,06	$5,00 \times 10^{-4}$	5	93,60
- 300	2000	1700	6,89	$2,00 \times 10^{-4}$	10	171,94
- 300	2600	2300	7,04	$1,67 \times 10^{-4}$	20	176,67
- 750	2250	1500	5,79	$1,87 \times 10^{-4}$	10	102,64
+ 300	1400	1700	6,11	$5,00 \times 10^{-4}$	10	75,82
+ 300	1700	2000	6,43	$4,00 \times 10^{-4}$	10	44,11
+ 600	2000	2600	6,59	$2,00 \times 10^{-4}$	10	73,98
+ 750	2250	3000	5,89	$1,87 \times 10^{-4}$	10	110,70
Valores Médios				$2,83 \times 10^{-4}$	10	106,01
Valores Adotados				$2,80 \times 10^{-4}$	10	106,00

Outros parâmetros necessários para construção do controlador em QDMC são o horizonte do modelo (N), o horizonte de controle (L), o horizonte de

predição (R), fator de ponderação para os movimentos de controle (f), fator de ponderação para diferença predita (q) e o tempo de controle (k).

O horizonte de controle, segundo Aström e Wittenmark (1989), é satisfatório considerá-lo igual a 1 ($L=1$) para sistemas simples como o sistema em estudo. Seborg et al. (1989) consideram que o horizonte de predição mínimo a ser considerado, deve ser proporcional ao tempo de ascensão da variável controlada. Como existe restrição pelo controlador, o fator de ponderação para os movimentos de controle foi reduzido a um valor mínimo, visto que não pode ser zero. Já para a ponderação da diferença predita não foi estabelecido fator.

8.2. Experimentos em malha fechada

Com o modelo matemático desenvolvido na identificação do processo em estudo foi possível desenvolver o controlador tipo QDMC. Uma das vantagens deste controlador é que é possível estabelecer limites de atuação para as variáveis manipuladas. Estes limites fazem parte do desenvolvimento do controlador, melhorando a estabilidade do controle dinâmico e evitando saturação dos sistemas envolvidos.

Como colocado anteriormente, os limites estabelecidos para a variável manipulada foram 1100 rpm como limite inferior, pois só a partir desta rotação que era possível verificar o borbulhamento no canal de aeração, e 3300 rpm com limite superior, pois tratava-se da maior rotação que o soprador era capaz de alcançar.

Foram realizados diversos testes, aplicando perturbações degrau no “*set-point*”, variações críticas no “*set-point*” e na vazão de água do canal, bem como observou-se a operação em malha fechada apenas com as perturbações naturais, ou seja, sem nenhuma perturbação premeditada. Foram realizados também experimentos com diferentes tempos de controle ou instante amostral

(k), operado com valores de 50, 100 e 150 segundos. As Figuras seguintes mostram os gráficos com os resultados dos experimentos realizados.

Para construção dos gráficos, foi utilizado a média móvel dos valores obtidos para concentração de oxigênio dissolvido de modo a suavizar picos e depressões agudos, visto que o sensor de oxigênio dissolvido sofre muita influência de microbolhas de ar que podem aderir a membrana sensorial. Isso possibilitou melhor visualização dos resultados e do comportamento da variável controlada.

Figura 25. Perturbação tipo degrau negativo no “set-point”; tempo de controle (k) de 100 segundos.

Figura 26. Perturbação tipo degrau positivo no “*set-point*”; tempo de controle (k) de 100 segundos.

As Figuras 25 e 26 mostram que as restrições na variável manipulada foram importantes para que não cessasse o borbulhamento de ar no canal de aeração, no caso do degrau negativo, e a atuação excessiva do soprador, no caso do degrau positivo. Nota-se ainda que a resposta ao degrau negativo mostrou-se mais aderente ao “*set-point*” estabelecido. Provavelmente, o degrau positivo no “*set-point*” impõe uma concentração de OD na água acima da concentração de saturação nas condições operacionais em que o sistema se encontrava.

No gráfico apresentado na Figura 26 foi observado que a concentração de oxigênio se aproxima da saturação, sendo que, mesmo com alta vazão de ar no canal (rotação elevada), não é possível dissolver mais oxigênio no meio líquido. Além disso, pode ter ocorrido um excesso na dosagem de hipossulfito de sódio, estabelecendo um ponto de equilíbrio entre consumo e dissolução de oxigênio mais baixo que o esperado inicialmente.

Figura 27. Perturbação crítica no “set-point”: degrau negativo seguido de degrau positivo; tempo de controle (k) de 50 segundos.

Figura 28. Perturbação crítica no “set-point”: degrau positivo seguido de degrau negativo; tempo de controle (k) de 100 segundos.

Figura 29. Perturbação crítica no “set-point”: degrau negativo seguido de degrau positivo; tempo de controle (k) de 150 segundos.

Observando as Figuras 27, 28 e 29, novamente percebe-se que o processo responde melhor ao degrau negativo, se ajustando mais rapidamente ao “*set-point*” quando nessa situação. Fica claro também a importância das restrições na variável manipulada, evitando situações críticas já mencionadas. Agora é possível notar também a atuação dos fatores de ponderação na variável manipulada que o controlador permite estabelecer, evitando mudanças bruscas na rotação do soprador. Esses fatores de ponderação fazendo com que não ocorram aumentos súbitos na rotação do soprador, tampouco reduções também bruscas na variável manipulada. Isso faz que o sistema seja mais estável e consistente.

Os gráficos apresentados nas Figuras 27, 28 e 29 evidenciam como a resposta ao degrau negativo é mais rápida e aderente ao “*set-point*” do que a resposta ao degrau positivo. O controlador mostrou maior rapidez de aderência ao “*set-point*” ao degrau negativo nos três casos. Vale ressaltar que em ambos, o controlador alcançou o ponto de trabalho estabelecido. É notável também diferenças no comportamento da variável controlada com relação ao tempo de controle (k) adotado.

Outro fato notável nos gráficos apresentados nas Figuras 27, 28 e 29 é o comportamento da rotação do soprador quando de movimentos bruscos no “*set-point*”. O controlador impede que a variável manipulada receba comandos com grandes movimentos, o que poderia proporcionar oscilações da variável controlada, ajustando de forma suave a concentração de OD ao “*set-point*” estabelecido.

No experimento realizado para obtenção dos dados da Figura 29, houve a necessidade de interrupção do experimento por motivos de logística, porém o ajuste ao “*set-point*” ocorreria, com o mesmo comportamento observado nos experimentos das Figuras 27 e 28.

Figura 30. Perturbação crítica na vazão de água do canal de aeração; tempo de controle (k) de 50 segundos.

Figura 31. Perturbação crítica na vazão de água do canal de aeração; tempo de controle (k) de 100 segundos.

Figura 32. Perturbação crítica na vazão de água do canal de aeração; tempo de controle (k) de 150 segundos.

As Figuras 30, 31 e 32 mostram situações em que a perturbação aplicada ao processo é na vazão de água que entra no canal. Nos três casos apresentados, observa-se que em situações em que a vazão de água no canal é bastante discrepante, o controlador mantém a concentração de oxigênio dissolvido bastante aderente ao “*set-point*” estabelecido. Ainda sim, quando a vazão de água é aumentada para valores acima dos anteriormente trabalhados, o controlador continua mantendo a concentração desejada de oxigênio.

O comportamento observado nos gráficos das Figuras 30, 31 e 32 são muito semelhantes, mostrando que o controlador conseguiu boa aderência da variável controlada ao “*set-point*” proposto.

Na Figura 30, o experimento partiu de um ponto onde a concentração de OD ainda não estava ajustada ao “*set-point*”. Isso prejudica a análise do comportamento da variável controlada quando do degrau negativo na vazão de água no canal. Esperava-se um pequeno aumento da concentração de OD imeditamente após a perturbação, porém observa-se uma inércia no comportamento do sistema que impede a verificação desse padrão de comportamento, como nos demais experimentos e mesmo neste, quando do degrau positivo na vazão de água, quando esperasse ligeira redução da concentração de OD na água.

Figura 33. Operação em malha fechada sem perturbação premeditada; tempo de controle (k) de 100 segundos.

A Figura 33 mostra que mesmo sem nenhuma perturbação em parâmetros possíveis de serem modificados (“*set-point*” e vazão de água), existem perturbações inerentes do processo que também impactam na operação do sistema de aeração. Ainda assim, o controlador também atua diante destas perturbações.

A instabilidade verificada nos gráficos apresentados pode ser explicada em parte pelas dificuldades enfrentadas na dosagem de hipossulfito de sódio durante a realização dos experimentos. Tratava-se de um sistema de gotejamento manual e não havia controle da concentração da solução que era dosada.

Observa-se nos gráficos apresentados a resiliência do processo de aeração em meios líquidos. Isso reforça a importância de desenvolver controladores avançados para estes sistemas buscando o máximo de otimização. Mostra também a importância do controle preditivo, visto a elevado tempo de resposta aos estímulos.

9. CONCLUSÕES

A partir dos dados gerados quando da operação do sistema de aeração em malha aberta para identificação do modelo de linear de primeira ordem que representasse adequadamente o processo em estudo, é possível inferir que o processo tem comportamento típico de sistemas não-lineares, ou seja, seu comportamento é diferenciado quando a concentração de OD está aumentando e quando está reduzindo.

Fazendo uma observação geral dos dados gerados nas operações em malha fechada, é possível concluir que o processo de aeração em estudo possui uma inércia considerável. Isto porque, em alguns casos, não é possível notar na variável controlada (concentração de OD) a perturbação imposta ou no “*set-point*” ou na vazão de água, continuando a tendência que se mostrava anteriormente. Além disso, mesmo quando a variável controlada está bem ajustada ao “*set-point*”, é bastante sutil o movimento desta variável quando da ocorrência de perturbações.

Observando os dados gerados pela operação em malha fechada, o controlador mostrou boa aderência ao “*set-point*” em praticamente todos os experimentos. Apenas dois deles (Figuras 26 e 29) não ocorreu o ajuste adequadamente.

A possibilidade de restringir a variável manipulada foi importante para o sistema de aeração tanto por limites físicos dos equipamentos (verificados nos resultados) quanto também quando houver limitações do processo em que será aplicado o controle.

Mesmo diante de perturbações não controladas, provavelmente relacionadas com o composto utilizados para remoção do OD e ao próprio sensor de OD, o controlador reagiu mostrando boa condição de atuação.

Na comparação dos tempos de controle testados, o tempo de controle de 50 segundos é melhor que os demais, visto o tempo em que alcança o “*set-point*”

estabelecido e o bom ajuste quando da ocorrência da perturbação, mostrando bom desempenho para o processo em estudo.

O controlador avançado do tipo QDMC mostrou-se adequado para aplicação em sistemas de aeração por ar difuso em canais de água, visto: a boa aderência ao “*set-point*”, ajustando se bem as diversas condições estabelecidas; a possibilidade de restrições na variável manipulada, preservando os equipamentos relacionados; a atuação diante de perturbações não previstas, ou seja, aquelas não controláveis; e pela sua robustez, mesmo diante de um processo com não-linearidades e de respostas lentas quando da aplicação das perturbações. É viável, então, realizar estudos para sua aplicação em sistema de tratamento de efluentes, visto que estes operam em condições semelhantes às estabelecidas neste estudo.

10. RECOMENDAÇÕES

Pelas observações realizadas durante a operação do sistema de aeração, o padrão de bolhas geradas pelos difusores deve ser melhorado de modo a obter bolhas com diâmetros menores, para melhorar a transferência de massa da bolha para o meio líquido, e distribuir melhor as bolhas longitudinalmente no canal.

É importante que o sistema de aeração seja dimensionado já de modo a otimizar o processo de transferência de massa de oxigênio da bolha de ar para o meio líquido. Isso implicará em melhores condições operacionais para o sistema e conseqüente um ajuste mais fino da concentração de OD, melhorando o desempenho do controlador.

Para contornar a influência de outras variáveis na concentração de OD, é interessante desenvolver controladores multivariáveis, estudando a influência de parâmetros como pH e potencial redox na variável controlada proposta.

A realização de estudos em sistemas de tratamento operando é interessante para o desenvolvimento dos controladores. Isso possibilitará observar melhor o comportamento das variáveis envolvidas quando da operação em condições reais dos reatores biológicos. Além disso, o consumo de OD em sistemas biológicos aeróbios é intensa o que modificará bastante o sistema quando das respostas as perturbações do sistema.

Assim, as principais recomendações para os próximos estudos são:

- Implementação de controle multivariável;
- Aplicação em reator biológico em operação (escala piloto ou real);
- Dimensionar o sistema de aeração de modo a otimizar a transferência de massa gás-líquido;
- Distribuir mais adequadamente os difusores pelo canal.

11. **REFERÊNCIAS BIBLIOGRÁFICAS**

ALMEIDA NETO, E.; RODRIGUES, M.A.; ODLOAK, D. (2000). **Robust Predictive Control of a Gasoline Debutanizer Column**. Brazilian Journal of Chemical Engineering, vol. 17, n. 4-7, p.967-977, 2000.

ANDREWS, J.F. (1974). **Dynamic models and control strategies for wastewater treatment processes**. Water Research, vol. 8, n. 5, p. 261-289, May 1974..

ASTRÖM, K.J.; WITTENMARK, B. (1989). **Adaptive Control**. P. 526, Addison-Wesley, Reading, Mass.

BAILEY, J.E.; OLLIS, D.F. (1973). **Biochemical Engineering Fundamentals**. 2nd edition. McGraw-Hill, New York, 1973.

BARBOSA JÚNIOR, A. R.. (1989). **Desenvolvimento de metodologia para a determinação do coeficiente de reaeração dos escoamentos naturais da água com o emprego de traçador gasoso**. Dissertação de Mestrado, Departamento de Hidráulica e Saneamento, Escola de Engenharia de São Carlos, Universidade de São Paulo, USP, 1989.

BEAL, L.L.; MONTEGGIA, L.O.; GIUSTINA, S.V.D. (2006). **Otimização de uma estação de tratamento de efluentes de uma indústria de embalagens de papel**. Revista Engenharia Sanitária e Ambiental, vol. 11, n. 3, p. 283-289, Julho/Setembro de 2006.

CARAMAN, S.; SBARCIOG, M.; BARBU, M. (2007). **Predictive Control of a Wastewater Treatment Process**. International Journal of Computers, Communications & Control, vol. II, n. 2, p. 123-142, 2007.

CHACHUAT, B.; ROCHE, N.; LATIFI, M.A. (2005). **Optimal aeration control of industrial alternating activated sludge plants**. Biochemical Engineering Journal, vol. 23, p. 277-289, January 2005.

CHOTKOWSKI, W.; BRDYS, M.A.; KONARCZARK, K. (2005). **Dissolved oxygen control for activated sludge processes**. International Journal of Systems Science, vol. 36, n. 12, p. 727-736, October 2005.

CORRÊA, L. C. (2006). **Controle do processo da transferência de oxigênio em corpos hídricos**. Dissertação (Mestrado em Ciências da Engenharia Ambiental). Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2006. 95p.

CORRÊA, N. A. (2000). **Projeto de Parâmetros e Tratamento de Restrições em Controlador Preditivo DMC**. Dissertação (Mestrado em Engenharia Química). Departamento de Engenharia Química, Universidade Federal de São Carlos, São Carlos, 2000. 103p.

CORRÊA, N. A.; FREIRE, F. B.; CORRÊA, R. G.; FREIRE, J. T. (2004). **Industrial Trials of Paste Drying in Spouted Beds Under QDMC**. Drying Technology, vol. 22, n. 5, p. 1087-1105, 2004.

CORRÊA, N. A.; FREIRE, F. B.; CORRÊA, R. G.; FREIRE, J. T. (2006). **Comparison among PI, DMC, QDMC, and GPC algorithms in the control of the spouted bed dryer**. Chemical Engineering Communications, vol. 193, n. 1, p. 24-37, January 2006.

COSTA, F.C.; RODRIGUES, F.A.M.; FONTOURA, G.A.T.; CAMPOS, J.C.; SANT'ANNA JÚNIOR, G.L.; DEZOTTI, M. (2003). **Tratamento do efluente de uma indústria química pelo processo de lodos ativados convencional e combinado com carvão ativado**. Revista Engenharia Sanitária e Ambiental, vol. 8, n. 4, p. 274-284, Outubro/Dezembro de 2003.

CUTLER, C. R.; RAMAKER, B. L. (1980). **Dynamic matrix control – a computer control algorithm**. In Proceedings of the joint automatic control conference.

FAZOLO, A. (2003). **Determinação de parâmetros cinéticos e de transferência de massa em reator radial aeróbio-anóxico alimentado com esgoto sanitário tratado em reator anaeróbio**. Tese de Doutorado em Engenharia Hidráulica e Saneamento, Escola de Engenharia de São Carlos, Universidade de São Paulo, USP, 2003.

GARCIA, C.E.; MORSHEDI, A.M. (1986). **Quadratic Programming Solution of Dynamic Matrix Control (QDMC)**. Chem. Eng. Commun., vol. 46, n. 73. 1986.

GARCÍA, C.E.; PRETT, D.M.; MORARI, M. (1989). **Model Predictive Control: Theory and Practice – a Survey**. Automatica, vol. 25, n. 3, p. 335-348, 1989.

HAMILTON, R.; BRAUN, B.; DARE, R.; KOOPMAN, B.; SVORONOS, S.A. (2006). **Control issues and challenges in wastewater treatment plants**. IEEE Control Systems Magazine, vol. 26, n. 4, p. 63-69, August 2006.

HIGBIE, R. (1935). **The rate of absorption of a pure gas into a still liquid during short periods of exposure**. Journal of American Inst. Chem. Engineers Trans, vol. 31, p. 365-390, 1935.

HOLENDA, B.; DOMOKOS, E.; RÉDEY, Á.; FAZAKAS, J. (2008). **Dissolved oxygen control of the activated sludge wastewater treatment process using model predictive control**. Computers and Chemical Engineering, vol. 32, p. 1270-1278, 2008.

LEE, B.K.; SUNG, S.W.; CHUN, H.D.; KOO, J.K. (1998). **Automatic Control for DO and pH in the Activated Sludge Process in a Coke Wastewater Treatment Plant**. Water Science Technology, vol. 37, n. 12, p. 141-148, 1998.

MARCHETTI, J.L.; MELLICHAMP, D.A.; SEBORG, D.E. (1983). **Predictive Control Based on Discrete Convolution Models**. Industrial & Engineering Chemistry Process Design and Development, vol. 22, n. 3, p. 488-495, 1983.

METCALF & EDDY (2003). **Wastewater Engineering: treatment and reuse**. Metcalf & Eddy, Inc. 4 ed., 1819 p., 2003.

ÖSKAN, G.; ÜSER, A.; HAPOGLU, H.; ALPHAZ, M. (2008). **Experimental application of linear/nonlinear dynamic matrix control to a reactor of limestone slurry titrated with sulfuric acid**. Chemical Engineering Journal, vol. 137, p. 320-327, 2008.

QIN, S.J.; BADGWELL, T.A. (2003). **A survey of industrial model predictive control technology**. Control Engineering Practice, vol. 11, n. 7, p. 733-764, July 2003.

RICKER, N.L. (1985). **Use of Quadratic Programming for Constrained Internal Model Control**. Ind. Eng. Chem. Proc. Des. Dev., vol. 24, n. 4. 1985.

RODRIGO, M.A.; SECO, A.; FERRER, J.; PENYA-ROJA, J.M.; VALVERDE, J.L. (1999). **Nonlinear control of an activated sludge aeration process: use of fuzzy techniques for tuning PID controllers**. ISA Transactions, vol. 38, n. 3, p. 231-241, July 1999.

RODRIGUES, J.A.D.; MACIEL FILHO, R. (1999). **Production optimization with operating constraints for a fed-batch reactor with DMC predictive control**. Chemical Engineering Science, vol. 54, n. 13-14, p. 2745-2751, July 1999.

SCHÜTZE, M.; CAMPISANO, A.; COLAS, H.; SCHILLING, W.; VANROLLEGHEM, P.A. (2004). **Real time control of urban wastewater systems—where do we stand today?** Journal of Hydrology, vol. 299, n. 3-4, p. 335-348, December 2004.

SCHROEDER, E. D. (1977). **Water and wastewater treatment**. McGraw-Hill, 1977, p.97.

SEBORG, D. E.; EDGAR, T. F.; MELLICHAMP, D. A. (1989). **Process Dynamics and Control**. Wiley, New York, 1989.

SHIAU, C.C. (1995). **Oxygen transfer in bubble and bubbleless aeration systems**. Ph D. thesis, University of Wollongong, Department of civil and Mining Engineering. 1995.

SPOSITO, V.A. (1975). **Linear and nonlinear programming**. Iowa State University Press, 269p., 1975.

STEPHANOPOULOS, G. (1983). **Chemical Process Control: An Introduction to Theory and Practice**. Prentice Hall, October 1983.

VISSMAN, JR.W.; HAMMER, M.J. (1998). **Water Supply and Pollution Control**, 6th ed. U.S.A., Addison-Wesley, 827p.

ZAFIRIOU, E.; MARCHAL, A.L. (1991). **Stability of SISO Quadratic Dynamic Matrix Control with Hard Output Constraints**. AIChE Journal, vol. 37, n. 10. 1991.

ZHANG, P.; YUAN, M.; WANG, H. (2006). **Self-tuning PID based on adaptive genetic algorithms with the application of activated sludge aeration**

process. Proceedings of the 6th World Congress on Intelligent Control and Automation, 21 to 23 June 2006, Dalian, China.

ANEXO I

Programação em linguagem C para operacionalização do controlador QDMC

```
#include <stdio.h>
#include <graph.h>
#include <conio.h>
#include <math.h>
#include <stdlib.h>
#include <dos.h>
#include <time.h>
#include <malloc.h>

/* ===== */
/* =   Programacao basica da placa e leitura por polling   = */
/* =   A placa e' lida no modo simples (nao burst)         = */
/* ===== */
/* ===== */
/* =   Rotina  IniciaEnderecoHardware                       = */
/* =                                                     = */
/* =   A placa ocupa 16 enderecos de entrada/saida        = */
/* = a partir do endereco base definido pelo con-        = */
/* = junto de JUMPER's J6.                                = */
/* =                                                     = */
/* ===== */

/* Endereco base da placa CAD12/36 */
#define EndBase 0x0380

int CadCtr0 ; /* Reg. Contador 0 da CAD12/36 */
int CadCtr1 ; /* Reg. Contador 1 da CAD12/36 */
int CadCtr2 ; /* Reg. Contador 2 da CAD12/36 */
int CadModo ; /* Reg. de Modo do Timer */
int CadStatus; /* Reg. Estado da CAD12/36 */
int ByteA ; /* Reg. Byte A do conversor A/D */
int ByteB ; /* Reg. Byte B do conversor A/D */
int CadEnd ; /* Reg. de Endereco */
int CadDado ; /* Reg. dado de escr. memoria */
int Divisor ;

void IniciaEnderecoHardware ()

{
    CadCtr0 = EndBase;
    CadCtr1 = EndBase + 1;
}
```

```

CadCtr2    = EndBase + 2;
CadModo    = EndBase + 3;
CadStatus  = EndBase + 3;
ByteA      = EndBase + 4;
ByteB      = EndBase + 5;
CadEnd     = EndBase + 4;
CadDado    = EndBase + 5;

} /* IniciaEnderecoHardware */

/* ===== */
/* =          Rotina  EscreveRegSecundario          = */
/* =          =                                     = */
/* =  A CAD12/36 possui um registro para  armazena- = */
/* =  mento de enderecamento para acesso de outros - = */
/* =  registradores internos da placa.              = */
/* =          =                                     = */
/* = ===== */

/* Enderecos secundarios */
#define SecLimite      0  /* Reg. de Limite */
#define SecPonteiro    1  /* Ponteiro Memoria */
#define SecComAD       2  /* Comando Conv. A/D */
#define SecRM          3  /* Reg. de Modo */
#define SecMemoria     4  /* Escrita Memoria */
#define SecLimpaFifo   5  /* Limpa Fifo */
#define SecAutoCal     6  /* Auto Calibracao */

void EscreveRegSecundario (char EndReg, char Dado)

{
    outp (CadEnd,EndReg);
    outp (CadDado,Dado);
} /* EscreveRegSecundario */

/* ===== */
/* =          Rotina  EscreveMemCanais              = */
/* =          =                                     = */
/* =  A CAD12/36 possui uma memoria de 16 posicoes = */
/* =  (0 a 15), que armazena o canal a ser converti- = */
/* =  do e a faixa de entrada deste canal.          = */
/* =          =                                     = */
/* =  Formato do dado armzazenado na memoria:        = */
/* =  D7 - 0 = unipolar, 1 = bipolar                 = */
/* =  D6 = 0 ativa ganho 100 (100mV)                 = */
/* =  D5 = 0 ativa ganho 5 (2 V)                     = */
/* =  D4 = 0 ativa ganho 2 (5 V)                     = */
/* =  D3 a D0 - canal                                = */
/* =          =                                     = */
/* =  OBS: - D6=D5=D4= 1, ativa ganho 1 (10V)        = */
/* =  - D3 e' o bit mais significativo do canal      = */
/* =  Na CAD12/36 REV.4 os dados sao invertidos     = */
/* =          =                                     = */
/* = ===== */

void EscreveMemCanais (char Posicao, char Dado)

{
    EscreveRegSecundario (SecPonteiro, Posicao);
    EscreveRegSecundario (SecMemoria, Dado);
} /* EscreveMemCanais */

```

```

/* ===== */
/* =          Rotina  AutoCalibracao          = */
/* =                                           = */
/* =  A auto-calibracao elimina os erros de nao- = */
/* = linearidade durante o processo de conversao. = */
/* = Deve ser executado no inicio do programa. = */
/* =                                           = */
/* ===== */

void AutoCalibracao ()

{
    int i,j;

    EscreveRegSecundario (SecAutoCal, 0);
    /* Aguarda pelo menos 800 ms */
    j = 800;
    do
    {
        i = 1000;
        do
            i--;
        while (i > 0);
        j--;
    } while (j > 0);
} /* AutoCalibracao */

/* ===== */
/* =          Rotina  LimpaFIFO          = */
/* =                                           = */
/* =  A CAD12/36 possui uma memoria FIFO (First In = */
/* = First Out) que armazena dados ja' convertidos = */
/* = e deve ser limpa a cada inicio de operacao. = */
/* = ate a R.4 a limpeza e' feita pela leitura da = */
/* = mesma, retirando os dados invalidos. = */
/* =                                           = */
/* =  Na revisao 6 basta escrever qualquer dado = */
/* = no endereco secundario no. 5 que sera resetada = */
/* = a Fifo = */
/* =                                           = */
/* ===== */

void LimpaFIFO ()
{
    int i, Dado;

    /* = for  (i = 1; i <= 17 ; i++)  valido ate a rev 4 = */
    /* = {                               = */
    /* =     inp (ByteA);                = */
    /* =     inp (ByteB);                = */
    /* = }                               = */

    EscreveRegSecundario (SecLimpaFifo, Dado);
} /* LimpaFIFO */

/* ===== */
/* =          Rotina  ProgramaMemoriaCanais          = */
/* =                                           = */
/* =  Carrega a memoria de canais com dados apro- = */
/* = priados para este exemplo. = */

```

```

/* =                                     = */
/* ===== */

/* constante para programar o ganho 1, 2, 5, 100      */
#define G1      0x70
#define G2      0x60
#define G5      0x50
#define G100    0x30

/* constante para programar tipo de faixa de entrada */
#define Bip     0x80
#define Unip    0x00

float Ganho[16];      /* tabela de ganhos dos canais      */
char Unipolar[16];    /* tabela indicativa de modo Unipolar */

void ProgramaMemoriaCanais ()

{
    EscreveRegSecundario (SecLimite, 0x0F); /* Carrega 15 no Reg. Limite */

    Ganho[0] = 1.0;
    Unipolar[0] = 0;
    EscreveMemCanais (0, Bip|G1|0); /* mem. 0, canal 0, bip. e ganho 1 */
    Ganho[1] = 1.0;
    Unipolar[1] = 0;
    EscreveMemCanais (1, Bip|G1|1); /* mem. 1, canal 1, bip. e ganho 1 */
    Ganho[2] = 1.0;
    Unipolar[2] = 0;
    EscreveMemCanais (2, Bip|G1|2); /* mem. 2, canal 2, bip. e ganho 1 */
    Ganho[3] = 1.0;
    Unipolar[3] = 0;
    EscreveMemCanais (3, Bip|G1|3); /* mem. 3, canal 3, bip. e ganho 1 */
    Ganho[4] = 1.0;
    Unipolar[4] = 0;
    EscreveMemCanais (4, Bip|G1|4); /* mem. 4, canal 4, bip. e ganho 1 */
    Ganho[5] = 1.0;
    Unipolar[5] = 0;
    EscreveMemCanais (5, Bip|G1|5); /* mem. 5, canal 5, bip. e ganho 1 */
    Ganho[6] = 1.0;
    Unipolar[6] = 0;
    EscreveMemCanais (6, Bip|G1|6); /* mem. 6, canal 6, bip. e ganho 1 */
    Ganho[7] = 1.0;
    Unipolar[7] = 1;
    EscreveMemCanais (7, Unip|G1|7); /* mem. 7, canal 7, bip. e ganho 1 */
    Ganho[8] = 1.0;
    Unipolar[8] = 1;
    EscreveMemCanais (8, Unip|G1|8); /* mem. 8, canal 8, unip. e ganho 1 */
    Ganho[9] = 1.0;
    Unipolar[9] = 1;

```



```

{
    int i;

    EscreveRegSecundario (SecComAD, Cn);    /* pede conversao */
    do
    {
        i= inp(CadStatus);
    } while ((i & 0x0001) == 0);
    i = inp(ByteA);    /* leia Byte A antes */
    i = i| (inp(ByteB)<<8);
    return ( i );

} /* LeConv */

/* ===== */
/* =                Funcao GerSin                = */
/* = Converte um sinal analogico (1-5v) no respectivo= */
/* = codigo para saida (complemento de dois).      = */
/* ===== */

int GerSin (float Sinal)
{
    float data;
    data=16384.0*2.0*((Sinal-5.0000))/5.0;
    /*printf("                %9.5f ",data);*/

    return(16384.0*0.4*(Sinal-5.0000));
} /* GerSin */

/* ===== */
/* =                Funcao EscSai                */
/* = Escreve dados nos canais de saida dos conversores */
/* = D/A                                          */
/* ===== */

int EscSai (int Conver, char Can)
{
    int SaiA, SaiB, ConvA, ConvB;
    SaiA=EndBase+8+(2*Can);
    SaiB=EndBase+8+(2*Can)+1;
    ConvA=Conver<<8;
    ConvA=ConvA>>8;
    ConvB=Conver>>8;
    outp(SaiA,ConvA);
    outp(SaiB,ConvB);
} /* EscSai */

/* ===== */
/* =                Funcao Tensao                = */
/* =                = */
/* = Converte leitura em valores de tensao.      = */
/* =                = */
/* = Os dados estao no formato complemento de 2   = */
/* = justificados a esquerda, variando de -32768 = */
/* = a 32752 em passos de 16.                    = */
/* =                = */
/* ===== */

float Tensao (int Leitura, char Canal)

```

```

{
    if (Unipolar[Canal] == 1)
        return (((5.0 * Leitura / 32768.0) + 5.0) / Ganho[Canal]);
    else
        return ((10.0 * Leitura) / (32768.0 * Ganho[Canal]));
} /* Tensao */

/* ===== */
/* =                      Rotina TelaApresentacao                      = */
/* ===== */

void TelaApresentacao ()

{
    int i;
    _GCLEARSCREEN;
    if (_setvideomode(_VRES16COLOR)==0)
        {printf("can't set mode");exit(1);}
    _setbkcolor(14);

    printf (" \n\n\n");
    printf ("  CONTROLADOR PREDITIVO MULTIVARIAVEL  \n");
    printf("\n\n  Aplica#o: Canal de Aera#o");
    printf("\n\n  Estrat,gia:");
    printf (" \n  Utiliza o algoritmo QDMC\n");
    printf ("\n\n\n  Tiago M. Soares e Nivaldo A. Correa\n \n");
    printf ("      SHS-EESC-USP,  PPG-SHS,  Sao Carlos  -  SP,  2008\n\n\n\n\n\n\n\n\n\n\n");
    printf ("  Digite 'i'(ENTER) para iniciar.");
    for(;;){if(getchar()=='i'){break;}}
    for (i=0;i<1;i++)
        printf("\n");

    _setvideomode(_DEFAULTMODE);
} /* TelaApresentacao */

/* ===== */
/* =                      Funcao Atraso                      */
/* =  Atrasa o fluxo de programa em T segundos.                      */
/* ===== */

void Atraso(float T)
{
    long int i;
    float tim;
    tim=clock();
    for(;;){
        if(kbhit()){break;}
        if(clock())>=tim+T*1000.0){break;}
    }
} /* Atraso */

/* ===== */
/* =                      Funcao rota                      = */
/* ===== */

float rota(float rot,char CanS)
{
    int Conver,LC;
    float Tesp,Sin;
    Sin=(10.0/3500.0)*rot;

```

```

if(Sin>9.999999){Sin=9.999999;}
Conver=GerSin(Sin);
/* printf("          %d \n",Conver);*/

EscSai(Conver,CanS);

Tesp=0.0;
Atraso(Tesp);
}

/* ===== */
/* =          Funcao oxdiss          = */
/* =          Leitura do Oxigênio Dissolvido          = */
/* ===== */
float oxdiss(char Can)
{
    int LC;
    float V,od;
    LC = LeConv (Can);
    V = Tensao (LC,Can);
    od=3.75*(V-1.0);
    od=0.03008+1.00679*od;
    /*printf(" %8.4f Volts <=> %8.4f ppm\n",V,od);*/

    return(od);
} /* umida */

/* ===== */
/* =          Funcao temp          = */
/* =          Leitura de temperatura          = */
/* ===== */
float temp(char Can)
{
    int LC;
    float V,T;
    LC = LeConv (Can);
    V = Tensao (LC,Can);
    T=12.5*(V-1.0);
    /* printf("          %f\n",V);*/
    return(T);
} /* temp */

/* ===== */
/* =          Funcao vaza          = */
/* =          Leitura de vazao de ar          = */
/* ===== */
float vaza(char Can)
{
    int LC;
    double V,DP,Q;
    LC = LeConv (Can);
    V = Tensao (LC,Can);
    DP=1.15654*V-0.16171;
    if(DP<=0.0){DP=0.0;}
    Q=10.0*(0.00002917)+0.01466*sqrt(DP);
    return(V);
} /* vaza */

```



```

/* ===== */
/* =          Funcao graf          = */
/* = Mostra o comportamento grafico de variaveis = */
/* ===== */

int graf(int num,float var,int tempo,float max,float min,int an)
{
    int a,b,c,d,n;
    long int i,j,m;
    float x,xx;

    if(num==1){a=45;b=15;c=215;d=115;}
    if(num==2){a=45;b=130;c=215;d=230;}
    if(num==3){a=45;b=245;c=215;d=345;}
    if(num==4){a=45;b=360;c=215;d=460;}

    if(num==5){a=250;b=15;c=420;d=115;}
    if(num==6){a=250;b=130;c=420;d=230;}
    if(num==7){a=250;b=245;c=420;d=345;}
    if(num==8){a=250;b=360;c=420;d=460;}

    if(tempo>=c-a+30-1){
        _setcolor(3);
        _rectangle(_GFILLINTERIOR,a,b,c,d);
        for (i=1;i<21;i++){
            j=i*5;
            _setcolor(1);
            if(i==5){_setcolor(0);}
            if(i==10){_setcolor(0);}
            if(i==15){_setcolor(0);}

            _moveto(a,b+j);
            _lineto(c,b+j);
        }
        for (i=1;i<35;i++){
            j=i*5;
            _moveto(a+j,b);
            _lineto(a+j,d);
        }
        _setcolor(14);
    }

    x=b+(var-min)*(d-b)/(max-min);
    xx=floor(x);
    if((x-xx)>=0.5){
        n=xx+1.0;
    }
    else {
        n=x;
    }
    if(an==-1){_moveto(a,d);}
    else{_moveto(a-30+tempo-1,d+b-an); _lineto(a-30+tempo,d+b-n);}

    return(n);
} /* graf */

/*****
/* PARAMETROS GERAIS DO CONTROLADOR E DECLARACAO DE FUNCOES */
/*****
#define NVC 2
#define NVM 2

```

```

#define R 7
#define L 1
#define NN 80

#define n (L*NVM) /* numero de variaveis (manipuladas) */
/* #define m (3*L*NVM+2*R*NVC) */ /* numero de restricoes (Du,u,y) */
#define m (3*L*NVM) /* numero de restricoes (Du,u) */
/* #define m (L*NVM) */ /* numero de restricoes (Du) */

void control(float K[NVC][NVM],float tal[NVC][NVM],
            float teta[NVC][NVM],float ym[NVC*NVM][NN],
            float D[R][NVC*NVM],float h[NN][NVC*NVM],float C[R],
            float B[R][L],float A[NVC*R][NVM*L],float fator[NVM],
            float alfa[NVC],float q[NVC],float r[NVC],float Dumax[NVM],
            float Dumin[NVM],float umax[NVM],float umin[NVM],
            float W[NVM*L][NVM*L],float QW[NVC*R][NVC*R],
            float AT[NVM*L][NVC*R],float S1[NVM*L][NVC*R],
            float S2[NVM*L][NVC*R],float S3[NVM*L][NVM*L],
            float c[NVM*L][NVM*L],float Dumaxim[1][NVM*L],
            float met2p[1][NVM*L],float fat2p[NVC*R][NVM*L],
            float FC[R][1],float a[3*L*(n+m+1)][L*NVM],float b[m+1][1],
            float Du[NVM][NN],float y[NVC],float u[NVM],
            float S[NVC][R],float TERM1[NVC][NVM],
            float TERM2[NVC][NVM],float P[NVC][R],float EE[NVC],
            float alf[R][NVC],float FX[R],
            float FCE[NVC*R],float E[NVC*R],
            float p[NVM*L][1],float X[NVM][1]);

void qp(float X[NVM][1],float c[NVM*L][NVM*L],
        float p[NVM*L][1],float a[3*L*(n+m+1)][L*NVM],
        float b[m+1][1],int pos[n+m+1],int pbase[n+m+1],float
E[n+m+1][2*(n+m)+1],
        float F[n+m+1][1],float P[n+m+1][3*(n+m)+1],float
custo[3*(n+m)+1],
        float base[n+m+1],float z[3*(n+m)+1],float zc[3*(n+m)+1],
        float zcc[3*(n+m)+1],float teta[n+m+1],float
Ps[n+m+1][3*(n+m)+1]);

void monitor(int *kcont,int *ind,float ods,float ts,float Q,
            float te,float rot,float ode,float odssp,int *an1,
            int *an2,int *an3,int *an4,int *an5,int *an6,int *an7,int
*an8);

void display(int k,float ts,float ods,float rot,float Q,float te,float
ode,float odssp);

void telagrafica(void);

int menui(int *TC,float *odssp);

void telagrafica(void);

/* ===== */
/* = Programa Principal = */
/* ===== */

main ()
{
    float *K,*tal,*teta,*ym,*D,*h,*C,*B,*A;
    float *fator,*q,*alfa,*r,*Dumax,*Dumin,*umax,*umin,*Dumaxim;
    float *W,*QW,*AT,*S1,*S2,*S3,*c;

```

```

float *met2p,*fat2p,*FC,*a,*b,*p;
float *Du,*y,*u,*X;
float *S,*TERM1,*TERM2,*P,*EE,*alf,*FX,*FCE,*E;

K=calloc(NVC*NVM,sizeof(float));
if(!K){printf("erro de alocacao 1");exit(1);}
tal=calloc(NVC*NVM,sizeof(float));
if(!tal){printf("erro de alocacao 2");exit(1);}
teta=calloc(NVC*NVM,sizeof(float));
if(!teta){printf("erro de alocacao 3");exit(1);}
ym=calloc(NVC*NVM*NN,sizeof(float));
if(!ym){printf("erro de alocacao 4");exit(1);}
D=calloc(R*NVC*NVM,sizeof(float));
if(!D){printf("erro de alocacao 5");exit(1);}
h=calloc(NN*NVC*NVM,sizeof(float));
if(!h){printf("erro de alocacao 6");exit(1);}
C=calloc(R,sizeof(float));
if(!C){printf("erro de alocacao 7");exit(1);}
B=calloc(R*L,sizeof(float));
if(!B){printf("erro de alocacao 8");exit(1);}
A=calloc(NVC*R*NVM*L,sizeof(float));
if(!A){printf("erro de alocacao 9");exit(1);}
q=calloc(NVC,sizeof(float));
if(!q){printf("erro de alocacao 10");exit(1);}
fator=calloc(NVM,sizeof(float));
if(!fator){printf("erro de alocacao 11");exit(1);}
alfa=calloc(NVC,sizeof(float));
if(!alfa){printf("erro de alocacao 12");exit(1);}
r=calloc(NVC,sizeof(float));
if(!r){printf("erro de alocacao 13");exit(1);}
Dumax=calloc(NVC,sizeof(float));
if(!Dumax){printf("erro de alocacao 14");exit(1);}
Dumin=calloc(NVC,sizeof(float));
if(!Dumin){printf("erro de alocacao 16");exit(1);}
umax=calloc(NVC,sizeof(float));
if(!umax){printf("erro de alocacao 17");exit(1);}
umin=calloc(NVC,sizeof(float));
if(!umin){printf("erro de alocacao 18");exit(1);}
W=calloc(NVM*L*NVM*L,sizeof(float));
if(!W){printf("erro de alocacao 19");exit(1);}
QW=calloc(NVC*R*NVC*R,sizeof(float));
if(!QW){printf("erro de alocacao 20");exit(1);}
AT=calloc(NVM*L*NVC*R,sizeof(float));
if(!AT){printf("erro de alocacao 21");exit(1);}
S1=calloc(NVM*L*NVC*R,sizeof(float));
if(!S1){printf("erro de alocacao 22");exit(1);}
S2=calloc(NVM*L*NVC*R,sizeof(float));
if(!S2){printf("erro de alocacao 23");exit(1);}
S3=calloc(NVM*L*NVM*L,sizeof(float));
if(!S3){printf("erro de alocacao 24");exit(1);}
c=calloc(NVM*L*NVM*L,sizeof(float));
if(!c){printf("erro de alocacao 25");exit(1);}
Dumaxim=calloc(NVC*L,sizeof(float));
if(!Dumaxim){printf("erro de alocacao 26");exit(1);}
met2p=calloc(NVC*L,sizeof(float));
if(!met2p){printf("erro de alocacao 27");exit(1);}
fat2p=calloc(NVC*L*NVM*R,sizeof(float));
if(!fat2p){printf("erro de alocacao 28");exit(1);}
FC=calloc(R,sizeof(float));
if(!FC){printf("erro de alocacao 29");exit(1);}
a=calloc(3*L*(n+m+1)*L*NVM,sizeof(float));

```

```

if(!a){printf("erro de alocacao 30");exit(1);}
b=calloc(m+1,sizeof(float));
if(!b){printf("erro de alocacao 31");exit(1);}
Du=calloc(NN*NVM,sizeof(float));
if(!Du){printf("erro de alocacao 32");exit(1);}
y=calloc(NVC,sizeof(float));
if(!y){printf("erro de alocacao 33");exit(1);}
u=calloc(NVM,sizeof(float));
if(!u){printf("erro de alocacao 34");exit(1);}
S=calloc(NVC*R,sizeof(float));
if(!S){printf("erro de alocacao 35");exit(1);}
TERM1=calloc(NVC*NVM,sizeof(float));
if(!TERM1){printf("erro de alocacao 37");exit(1);}
TERM2=calloc(NVC*NVM,sizeof(float));
if(!TERM2){printf("erro de alocacao 38");exit(1);}
P=calloc(NVC*R,sizeof(float));
if(!P){printf("erro de alocacao 39");exit(1);}
EE=calloc(NVC,sizeof(float));
if(!EE){printf("erro de alocacao 40");exit(1);}
alf=calloc(R*NVC,sizeof(float));
if(!alf){printf("erro de alocacao 41");exit(1);}
FX=calloc(R,sizeof(float));
if(!FX){printf("erro de alocacao 42");exit(1);}
FCE=calloc(NVC*R,sizeof(float));
if(!FCE){printf("erro de alocacao 43");exit(1);}
E=calloc(NVC*R,sizeof(float));
if(!E){printf("erro de alocacao 44");exit(1);}
p=calloc(NVM*L,sizeof(float));
if(!p){printf("erro de alocacao 45");exit(1);}
X=calloc(NVM,sizeof(float));
if(!X){printf("erro de alocacao 46");exit(1);}

control(K,tal,teta,ym,D,h,C,B,A,fator,alfa,q,r,
        Dumax,Dumin,umax,umin,W,QW,AT,S1,S2,S3,c,
        Dumaxim,met2p,fat2p,FC,a,b,Du,y,u,S,TERM1,
        TERM2,P,EE,alf,FX,FCE,E,p,X);

}/*main*/

/*****
/***** FUNCOES *****/
/*****/

void control(float K[NVC][NVM],float tal[NVC][NVM],
            float teta[NVC][NVM],float ym[NVC*NVM][NN],
            float D[R][NVC*NVM],float h[NN][NVC*NVM],float C[R],
            float B[R][L],float A[NVC*R][NVM*L],float fator[NVM],
            float alfa[NVC],float q[NVC],float r[NVC],float Dumax[NVM],
            float Dumin[NVM],float umax[NVM],float umin[NVM],
            float W[NVM*L][NVM*L],float QW[NVC*R][NVC*R],
            float AT[NVM*L][NVC*R],float S1[NVM*L][NVC*R],
            float S2[NVM*L][NVC*R],float S3[NVM*L][NVM*L],
            float c[NVM*L][NVM*L],float Dumaxim[1][NVM*L],
            float met2p[1][NVM*L],float fat2p[NVC*R][NVM*L],
            float FC[R][1],float a[3*L*(n+m+1)][L*NVM],float b[m+1][1],
            float Du[NVM][NN],float y[NVC],float u[NVM],
            float S[NVC][R],float TERM1[NVC][NVM],
            float TERM2[NVC][NVM],float P[NVC][R],float EE[NVC],
            float alf[R][NVC],float FX[R],
            float FCE[NVC*R],float E[NVC*R],
            float p[NVM*L][1],float X[NVM][1])

```

```

{

/*****
/*geracao da matriz dinamica e dos h's - definicao dos parametros e
restricoes*/
*****/

    int i,j,k,l,nj,e,s,ma,na,kj,pa;
    int Tss,T,N,TC,TE;
    float yma,ymb,ym1,ym11,ym12,ym2,ym21,ym22,tela;

    int t,sai,ind,an1,an2,an3,an4,an5,an6,an7,an8,an9;
    float
y1max,y1min,y2max,y2min,y3max,y3min,y4max,y4min,y5max,y5min,y6max,y6mi
n,y7max,y7min,y8max,y8min,y9min,y9max;
    float y1,y2,y3,y4,y5,y6,y7,y8,y9;
    double Q;
    float
Tesp,ods,odssp,odsl,ode,tu,ts,ts0,tu0,te,tes,te0,ta,ta0,tau,tau0,frmol
,fr,fr0,fre,fre0,ps,rot;
    float qar;
    int corre,cont,con,J;
    float Ys,Yssp,Ysl,Pr,Prsp,Pr1,tel;
    int kcont,Nr,seme[1];
    int top,left,bot,right;

    char
CanSP,CanQ,CanPV,CanH,CanT,CanTs,CanTe,CanTa,CanTau,CanPO,CanW,xx10[80
];

    float Hw,H1,H2,Har,cpag,lambw,omeg,Tespe,beta;
    float ran,sizea;

    int *pos,*pbase,kjj;
    float *E1,*F,*P1,*custo,*base,*z,*zc,*zcc,*tetal,*Ps;
    struct dostime_t tim;
    _dos_gettime(&tim);

    pos=calloc(n+m+1,sizeof(int));
    if(!pos){printf("erro de alocação 47");exit(1);}
    pbase=calloc(n+m+1,sizeof(int));
    if(!pbase){printf("erro de alocação 48");exit(1);}
    E1=calloc((n+m+1)*(2*(n+m)+1),sizeof(float));
    if(!E1){printf("erro de alocação 49");exit(1);}
    F=calloc(n+m+1,sizeof(float));
    if(!F){printf("erro de alocação 50");exit(1);}
    P1=calloc((n+m+1)*(3*(n+m)+1),sizeof(float));
    if(!P1){printf("erro de alocação 51");exit(1);}
    custo=calloc(3*(n+m)+1,sizeof(float));
    if(!custo){printf("erro de alocação 52");exit(1);}
    base=calloc((n+m+1),sizeof(float));
    if(!base){printf("erro de alocação 53");exit(1);}
    z=calloc(3*(n+m)+1,sizeof(float));
    if(!z){printf("erro de alocação 54");exit(1);}
    zc=calloc(3*(n+m)+1,sizeof(float));
    if(!zc){printf("erro de alocação 55");exit(1);}
    zcc=calloc(3*(n+m)+1,sizeof(float));
    if(!zcc){printf("erro de alocação 56");exit(1);}
    tetal=calloc(n+m+1,sizeof(float));
    if(!tetal){printf("erro de alocação 57");exit(1);}
    Ps=calloc((n+m+1)*(3*(n+m)+1),sizeof(float));

```

[illegible]

```

top=2;
left=90;
bot=3;
right=57;
an1=an2=an3=an4=an5=an6=an7=an8=an9=-1;
ind=0;
kcont=0;
sai=0;
Tespe=250.0;
ts0=temp(CanTs);
te0=temp(CanTe);
beta=0.2;

Dumax[0]=150.0;
Dumin[0]=-150.0;
Dumax[1]=50.0;
Dumin[1]=-50.0;

umax[0]=1200.0;
umin[0]=-1000.0;
umax[1]=400.0;
umin[1]=-400.0;

u[0]=0.0;
u[1]=0.0;

for (j=0;j<NN;j++){
    for (i=0;i<NVM;i++){
        Du[i][j]=0.0;
    }
}

K[0][0]=0.00028;
K[0][1]=0.00028;
K[1][0]=0.00028;
K[1][1]=0.00028;

tal[0][0]=106.0;
tal[0][1]=106.0;
tal[1][0]=106.0;
tal[1][1]=106.0;

teta[0][0]=10.0;
teta[0][1]=10.0;
teta[1][0]=10.0;
teta[1][1]=10.0;

for (i=0;i<(N+1);i++){
    tel=T*i;
    for (k=0;k<NVM;k++){
        for (j=0;j<NVC;j++){
            nj=j+k*NVC;
            if (tel<=teta[j][k]){
                ym[nj][i]=0.0;
            }
            if (tel>teta[j][k]){
                ym[nj][i]=K[j][k]*(1.0-exp(-1.0*(tel-
teta[j][k])/tal[j][k]));
            }
        }
    }
}

```

```

    }
}

for (k=0;k<NVM;k++) {
    for (j=0;j<NVC;j++) {
        nj=j+k*NVC;
        for (i=1;i<(R+1);i++) {
            D[i-1][nj]=ym[nj][i];
        }
        for (i=1;i<(N+1);i++) {
            h[i-1][nj]=ym[nj][i]-ym[nj][i-1];
        }
    }
}

for (j=0;j<NVC;j++) {
    for (k=0;k<NVM;k++) {
        nj=j+k*NVC;
        for (i=0;i<R;i++) {
            C[i]=D[i][nj];
        }
        for (l=0;l<R;l++) {
            for (ma=0;ma<L;ma++) {
                if (l>=ma) {
                    B[l][ma]=C[l-ma];
                }
                else {
                    B[l][ma]=0.0;
                }
            }
        }
        for (ma=0;ma<R;ma++) {
            for (na=0;na<L;na++) {
                A[j*R+ma][k*L+na]=B[ma][na];
            }
        }
    }
}

/*****
/*          matriz c e ponderacoes f e Q          */
*****/
for (k=0;k<NVM;k++) {
    for (i=0;i<L;i++) {
        for (j=0;j<L;j++) {
            W[k*L+i][k*L+j]=0;
            W[k*L+i][k*L+i]=fator[k];
        }
    }
}

for (i=0;i<NVC*R;i++) {
    for (j=0;j<NVC*R;j++) {
        QW[i][j]=0.0;
    }
}

for (s=0;s<NVC;s++) {
    for (j=0;j<R;j++) {
        l=s*R+j;
        QW[l][l]=q[s];
    }
}

```



```

for(i=0;i<NVC*R;i++){
    for(j=0;j<NVM*L;j++){
        AT[j][i]=A[i][j];
    }
}
for(i=0;i<NVM*L;i++){
    for(j=0;j<NVC*R;j++){
        S1[i][j]=0.0;
        for(k=0;k<NVC*R;k++){
            S1[i][j]=S1[i][j]+AT[i][k]*QW[k][j];
        }
    }
}
for(i=0;i<NVM*L;i++){
    for(j=0;j<NVC*R;j++){
        S2[i][j]=0.0;
        for(k=0;k<NVC*R;k++){
            S2[i][j]=S2[i][j]+S1[i][k]*QW[k][j];
        }
    }
}
for(i=0;i<NVM*L;i++){
    for(j=0;j<NVM*L;j++){
        S3[i][j]=0.0;
        for(k=0;k<NVC*R;k++){
            S3[i][j]=S3[i][j]+S2[i][k]*A[k][j];
        }
    }
}
for(i=0;i<NVM*L;i++){
    for(j=0;j<NVM*L;j++){
        c[i][j]=S3[i][j]+W[i][j];
    }
}

free(S1);
free(S2);
free(S3);
free(AT);

for(e=0;e<NVM*L;e++){
    for(j=0;j<L;j++){
        Dumaxim[0][e*L+j]=Dumax[e];
    }
}

for(j=0;j<NVM*L;j++){
    met2p[0][j]=0.0;
    for(k=0;k<NVM*L;k++){
        met2p[0][j]=met2p[0][j]+Dumaxim[0][k]*c[k][j];
    }
}

for(s=0;s<NVC;s++){
    for(j=0;j<R;j++){
        l=s*R+j;
        QW[l][l]=q[s]*q[s];
    }
}
for(i=0;i<NVC*R;i++){

```

```

        for (j=0; j<NVM*L; j++) {
            fat2p[i][j]=0.0;
            for (k=0; k<NVC*R; k++) {
                fat2p[i][j]=fat2p[i][j]+QW[i][k]*A[k][j];
            }
        }
    }

    free(W);
    free(QW);
    free(A);

    for (i=0; i<R; i++) {
        FC[i][0]=1.0;
    }

    /*****
    /*                                matriz de restricoes a                                */
    /*****/
    if (L==1) {
        a[0][0]=1.0;
        a[0][1]=0.0;
        a[1][0]=0.0;
        a[1][1]=1.0;
        a[2][0]=1.0;
        a[2][1]=0.0;
        a[3][0]=0.0;
        a[3][1]=1.0;
        a[4][0]=-1.0;
        a[4][1]=0.0;
        a[5][0]=0.0;
        a[5][1]=-1.0;
    }
    if (L==2) {
        a[0][0]=1.0;
        a[0][1]=0.0;
        a[0][2]=0.0;
        a[0][3]=0.0;
        a[1][0]=0.0;
        a[1][1]=1.0;
        a[1][2]=0.0;
        a[1][3]=0.0;
        a[2][0]=0.0;
        a[2][1]=0.0;
        a[2][2]=1.0;
        a[2][3]=0.0;
        a[3][0]=0.0;
        a[3][1]=0.0;
        a[3][2]=0.0;
        a[3][3]=1.0;
        a[4][0]=1.0;
        a[4][1]=0.0;
        a[4][2]=0.0;
        a[4][3]=0.0;
        a[5][0]=1.0;
        a[5][1]=1.0;
        a[5][2]=0.0;
        a[5][3]=0.0;
        a[6][0]=0.0;
        a[6][1]=0.0;
        a[6][2]=1.0;
    }

```

```

    a[6][3]=0.0;
    a[7][0]=0.0;
    a[7][1]=0.0;
    a[7][2]=1.0;
    a[7][3]=1.0;
    a[8][0]=-1.0;
    a[8][1]=0.0;
    a[8][2]=0.0;
    a[8][3]=0.0;
    a[9][0]=-1.0;
    a[9][1]=-1.0;
    a[9][2]=0.0;
    a[9][3]=0.0;
    a[10][0]=0.0;
    a[10][1]=0.0;
    a[10][2]=-1.0;
    a[10][3]=0.0;
    a[11][0]=0.0;
    a[11][1]=0.0;
    a[11][2]=-1.0;
    a[11][3]=-1.0;
}
/*****
/*                                     controlador                                     */
/*****/
for(i=0;i<NVC;i++){
    y[i]=0.0;
}
/*****/
con=0;
kjj=-1;
kj=0;
kcont=0;
for(;;){
    if(kbhit()){
        sai=menui(&TC,&odssp);
        kj=0;
    }
    if(sai==1){_setvideomode(_DEFAULTMODE);exit(0);}
    if(kcont>TE){

        kj=kj+1;
        if(kj==TC){
            kjj=kjj+1;
/*****/
/*                                     vetor dos limites b                                     */
/*****/
/* para os Du                                     */
/*----- */
        for(e=0;e<NVM;e++){
            for(i=0;i<L;i++){
                b[e*L+i][0]=Dumax[e]-Dumin[e];
            }
        }

/* para os u                                     */
/*----- */
        for(e=0;e<NVM;e++){
            for(i=0;i<L;i++){
                b[NVM*L+e*L+i][0]=(i+1)*Dumax[e]-umin[e]+u[e];
            }

```

```

}
for(e=0;e<NVM;e++){
    for(i=0;i<L;i++){
        b[2*NVM*L+e*L+i][0]=-1.0*(i+1)*Dumax[e]+umax[e]-u[e];
    }
}

/*****
/* controlador: calculo de S e P */
*****/
for(s=0;s<NVC;s++){
    for(ma=0;ma<R;ma++){
        S[s][ma]=0.0;
        for(e=0;e<NVM;e++){
            nj=s+e*NVC;
            TERM1[s][e]=0.0;
            TERM2[s][e]=0.0;
            for(i=(ma+1);i<N;i++){
                pa=N-1+ma-i;
                if(pa<=0){
                    TERM2[s][e]=0;
                }
                else{
                    pa=(N-1)+ma-i;
                    TERM1[s][e]=TERM1[s][e]+h[i][nj]*Du[e][pa];
                }
            }
            S[s][ma]=S[s][ma]+TERM1[s][e]+TERM2[s][e];
        }
    }
}
for(s=0;s<NVC;s++){
    P[s][1]=S[s][1];
    for(ma=0;ma<R;ma++){
        P[s][ma]=P[s][ma-1]+S[s][ma];
    }
}
for(e=0;e<NVM;e++){
    for(j=1;j<N;j++){
        Du[e][j-1]=Du[e][j];
    }
}

/*****
/* calculo do erro em malha aberta E e vetor de coef. linear p */
*****/
for(s=0;s<NVC;s++){
    EE[s]=r[s]-y[s];
    for(j=0;j<R;j++){
        alf[j][s]=alfa[s];
        FX[j]=(FC[j][1]-alf[j][s])*EE[s];
    }
    for(j=0;j<R;j++){
        FCE[s*R+j]=FX[j];
    }
}
for(s=0;s<NVC;s++){
    for(j=0;j<R;j++){
        l=s*R+j;
        E[l]=FCE[l]-P[s][j];
    }
}

```

```

    }

    for (j=0; j<NVM*L; j++) {
        p[j][0]=met2p[0][j];
        for (k=0; k<NVC*R; k++) {
            p[j][0]=p[j][0]-0.5*E[k]*fat2p[k][j];
        }
    }

/*****
/*      controlador: calculo dos movimentos Du      */
*****/

    qp(X,c,p,a,b,pos,pbase,E1,F,P1,custo,base,z,zc,zcc,tetal,Ps);

    for (e=0; e<NVM; e++) {
        Du[e][N-1]=Dumax[e]-X[e][0];
        u[e]=u[e]+Du[e][N-1];
    }

    kj=0;
}

/*****
/***** IMPLEMENTACAO, AQUISICAO E MONITORACAO GRAFICA *****/
*****/

/* VAR. MANIPULADAS */
/*-----*/

    rot=u[0]+2100.0;
/*      rot=1500.0; */
    rota(rot,CanPO);

/*-----*/
/* VAR. ASSOCIADAS */
/*-----*/

    te=temp(CanTs);
/*if((te-te0)>0.25){te=te0+0.25;}
if((te-te0)<-0.25){te=te0-0.25;}
te0=te; */
    ts=temp(CanTs);

    ode=oxdiss(CanP);

/*-----*/
/* VAR. CONTROLADAS */
/*-----*/

    ods=oxdiss(CanP);
/* ods=ods1+u[0]*0.00028*(1-exp(-(kcont-10.0)/106.0)); */
    Ys=Ys1+u[1]*0.00028*(1-exp(-(kcont-10.0)/106.0));

    Q=vaza(CanQ);

    y[0]=(ods-ods1);
    y[1]=(Ys-Ys1);

    r[0]=odssp-ods1;
    r[1]=Yssp-Ys1;

```

```

        /*if(kcont>8){odssp=6.3;}
        if(kcont>15){odssp=5.9;}*/

monitor(&kcont,&ind,ods,ts,Q,te,rot,ode,odssp,&an1,&an2,&an3,&an4,&an5
,&an6,&an7,&an8);
    }
    getch();
    _setvideomode(_DEFAULTMODE);

}/* control */
/*****
/*****

void qp(float x[NVM][1],float D[NVM*L][NVM*L],
        float c[NVM*L][1],float A[3*L*(n+m+1)][L*NVM],
        float b[m+1][1],int pos[n+m+1],int pbase[n+m+1],float
E[n+m+1][2*(n+m)+1],
        float F[n+m+1][1],float P[n+m+1][3*(n+m)+1],float
custo[3*(n+m)+1],
        float base[n+m+1],float z[3*(n+m)+1],float zc[3*(n+m)+1],
        float zcc[3*(n+m)+1],float teta[n+m+1],float
Ps[n+m+1][3*(n+m)+1])
{

    /* maximizar x'Dx+c'x */
    /* sujeito a: Ax<=b */
    /* x>=0 */

    /* n= no. de variaveis */
    /* m= no. de restricoes */

    int itmax,i,j,k,kk,jj,jjj,final,indica,indc,indl,resp,regra;
    float art,eps,menor,min;

    art=-1000.0;
    itmax=100; /* numero maximo de iteracoes */

    /* for(i=0;i<n+m;i++){
    for(j=0;j<NVM*L;j++){
        printf(" \n %f" ,A[i][j]);pausa();*/
    /*}
        printf("\n %d",i);
    } */

/*****
/* Arranjo do tableau P */
/*****
for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        E[i][j]=-1.0*D[i][j];
    }
    for(j=n;j<n+m;j++){
        E[i][j]=0.0;
    }
    for(j=n+m;j<n+2*m;j++){
        E[i][j]=-1.0*A[j-n-m][i];
    }
    for(j=n+2*m;j<2*(n+m);j++){
        E[i][j]=0.0;
        E[i][n+2*m+i]=1.0;
    }
}

```

```

}

for(i=n;i<n+m;i++){
    for(j=0;j<n;j++){
        E[i][j]=A[i-n][j];
    }
    for(j=n;j<n+m;j++){
        E[i][j]=0.0;
        E[i][i]=1.0;
    }
    for(j=n+m;j<n+2*m;j++){
        E[i][j]=0.0;
    }
    for(j=n+2*m;j<2*(n+m);j++){
        E[i][j]=0.0;
    }
}

for(i=0;i<n;i++){
    F[i][0]=-1.0*c[i][0];
    P[i][0]=F[i][0];
}
for(i=n;i<n+m;i++){
    F[i][0]=b[i-n][0];
    P[i][0]=F[i][0];
}
for(i=0;i<n+m;i++){
    for(j=1;j<2*(n+m)+1;j++){
        P[i][j]=E[i][j-1];
    }
}

/* for(i=0;i<m+n;i++){
    printf("  \n %f" ,P[i][2]);pausa();
}

    printf("\n ok"); */

/* anti ciclo */
/*----- */
eps=0.0000001;
for(i=0;i<(n+m);i++){
    P[i][0]=P[i][0]+eps;
    eps=eps*eps;
}
/* introducao das variaveis artificiais */
kk=0;
for(i=0;i<(n+m);i++){
    if(P[i][0]<0.0){
        for(j=0;j<(2*(n+m)+1);j++){
            P[i][j]=-1.0*P[i][j];
        }
    }
    pos[kk]=i;
    for(k=0;k<n+m;k++){
        P[k][2*(n+m)+1+kk]=0.0;
    }
    P[i][2*(n+m)+1+kk]=-1.0;
    if(P[i][0]<0){
        P[i][2*(n+m)+1+kk]=1.0;
    }
    kk=kk+1;
}

```

```

    }
    kk=kk-1;

/*for(j=0;j<27;j++){
for(i=0;i<m+n;i++){
    printf("  \n %f" ,P[i][j]);pausa();
}
printf("\n %d",j);
} */

/*****
/*definicao de custos, da base e das posicoes das variaveis na base */
*****/
for(j=0;j<(2*(n+m)+1);j++){
    custo[j]=0.0;
}
if(kk!=0){
    for(j=(2*(n+m)+1);j<(2*(n+m)+2+kk);j++){
        custo[j]=art;
    }
}

for(i=0;i<n;i++){
    base[i]=custo[n+2*m+1+i];
}
for(i=n;i<n+m;i++){
    base[i]=custo[i+1];
}
for(i=n;i<n+m;i++){
    pbase[i]=i+1;
}
for(i=0;i<n;i++){
    pbase[i]=n+2*m+1+i;
}

if(kk!=0){
    for(k=0;k<kk+1;k++){
        base[pos[k]]=custo[2*(n+m)+1+k];
        pbase[pos[k]]=2*(n+m)+1+k;
    }
}

final=0;
for(jj=0;jj<itmax;jj++){

/*****
/* calculo de z-c */
*****/
    for(i=0;i<2*(n+m)+2+kk;i++){
        z[i]=0.0;
        for(j=0;j<n+m;j++){
            z[i]=z[i]+base[j]*P[j][i];
        }
        zc[i]=z[i]-custo[i];
        zcc[i]=zc[i];
    }

/*****
/* determinacao do menor z-c negativo e verificacao da regra */
*****/
    indica=0;

```



```

indc=0;
resp=1;
jjj=0;
while(resp!=0){
    menor=0.0;
    for(i=1;i<2*(n+m)+1;i++){
        if(i!=indica){
            if(zcc[i]<menor){
                menor=zcc[i];
                indc=i;
            }
        }
    }
    if(jjj>=2*(n+m)){
        final=1;break;
    }
    if(indc==0){
        final=1;break;
    }
    if(indc>=1 && indc<=n){
        regra=n+2*m+indc;
    }
    if(indc>=(n+2*m+1) && indc<=2*(n+m)){
        regra=indc-n-2*m;
    }
    if(indc>=(n+1) && indc<=(n+m)){
        regra=indc+m;
    }
    if(indc>=(n+m+1) && indc<=(n+2*m)){
        regra=indc-m;
    }
    resp=0;
    for(i=0;i<m+n;i++){
        if(regra==pbase[i]){
            resp=resp+1;
            indica=indc;
            zcc[indc]=-1*zcc[indc];
        }
    }
    jjj=jjj+1;
}

/*    printf(" \n %d    %d",indc,regra);pausa();*/
if(final==0){
/*****
/* determinacao do menor teta positivo e definicao do pivot */
/*****/
    min=1.0e+10;
    for(i=0;i<n+m;i++){
        if(P[i][indc]!=0.0){
            teta[i]=P[i][0]/P[i][indc];
            if(teta[i]>=0.0){
                if(teta[i]<min){
                    min=teta[i];
                    indl=i;
                }
            }
        }
    }
}

/*****/

```

```

/* novo tableau */
/*****/
    for(j=0;j<2*(n+m)+kk+2;j++){
        for(i=0;i<n+m;i++){
            if(i==indl){
                Ps[i][j]=P[i][j]/P[indl][indc];
            }
            else{
                Ps[i][j]=P[i][j]-P[i][indc]*P[indl][j]/P[indl][indc];
            }
        }
    }
    for(j=0;j<2*(n+m)+kk+2;j++){
        for(i=0;i<n+m;i++){
            P[i][j]=Ps[i][j];
        }
    }

/*****/
/* troca de base */
/*****/
    base[indl]=custo[indc];
    pbase[indl]=indc;
}
else{

/*****/
/* solucao */
/*****/
    for(j=1;j<n+1;j++){
        for(i=0;i<(n+m);i++){
            if(pbase[i]==j){
                x[j-1][0]=P[i][0];
/*
                printf(" \n %f",x[j-1][0]);*/
            }
        }
        break;
    }
}
}/* qp */

void monitor(int *kcont,int *ind,float ods,float ts,float Q,float
te,float rot,float ode,float odssp,int *an1,
            int *an2,int *an3,int *an4,int *an5,int *an6,int *an7,int
*an8)
{
    FILE *arq1;
    int t,cont;
    float y1,y2,y3,y4,y5,y6,y7,y8,Tesp;
    float
y1max,y1min,y2max,y2min,y3max,y3min,y4max,y4min,y5max,y5min,y6max,y6mi
n,y7max,y7min,y8max,y8min;

    arq1=fopen("QDMCtms.dat","a");

    t=((*ind)+1)*30+(*kcont)-((*ind)*200);
    if(t>=200-1){*ind=*ind+1;}
    y1max=15.0;
    y2max=50.0;

```

```

y3max=3500;
y4max=200.0;
y5max=15.0;
y6max=50.0;
y7max=0.05;
y8max=0.1;

y1min=0.0;
y2min=0.0;
y3min=0.0;
y4min=0.0;
y5min=0.0;
y6min=0.0;
y7min=0.0;
y8min=0.0;

y1=ods;
*an1=graf(1,y1,t,y1max,y1min,*an1);
y2=ts;
*an2=graf(2,y2,t,y2max,y2min,*an2);
y3=rot;
*an3=graf(3,y3,t,y3max,y3min,*an3);
y4=100.0;
*an4=graf(4,y4,t,y4max,y4min,*an4);
y5=ode;
*an5=graf(5,y5,t,y5max,y5min,*an5);
y6=te;
*an6=graf(6,y6,t,y6max,y6min,*an6);
y7=Q;
if(y7>=y7max){y7=y7max;}
*an7=graf(7,y7,t,y7max,y7min,*an7);
y8=0.05;
*an8=graf(8,y8,t,y8max,y8min,*an8);
cont=(*kcont);
display(cont,ts,ods,rot,Q,te,ode,odssp);
Tesp=0.98;
Atraso(Tesp);
*kcont=*kcont+1;

fprintf(arq1,"%8.3f %8.2f %8.0f %8.2f \n",ods,odssp,rot,ts);
fclose(arq1);
}

void display(int k,float ts,float ods,float rot,float Q,float te,float
ode,float odssp)
{
char xx1[100];
int top,left,bot,right;
top=2;
left=90;
bot=3;
right=57;

    _settextwindow(top+4,left+16,bot+4,right+16);
    sprintf(xx1," %3.2f",ods);
    _settextcolor(7);
    _outtext(xx1);

    _settextwindow(top+5,left+16,bot+5,right+16);
    sprintf(xx1," %3.2f",10.0);
    _settextcolor(7);

```

```

_outtext(xx1);

_settextwindow(top+8, left+17, bot+8, right+17);
sprintf(xx1, " %4.0f", rot);
_settextcolor(7);
_outtext(xx1);

_settextwindow(top+9, left+16, bot+9, right+16);
sprintf(xx1, " %3.2f", 10.0);
_settextcolor(7);
_outtext(xx1);

_settextwindow(top+12, left+16, bot+12, right+16);
sprintf(xx1, " %3.3f", Q);
_settextcolor(7);
_outtext(xx1);

_settextwindow(top+13, left+16, bot+13, right+16);
sprintf(xx1, " %3.2f", ts);
_settextcolor(7);
_outtext(xx1);

/* _settextwindow(top+14, left+17, bot+14, right+17);
sprintf(xx1, " %3.2f", 10.0);
_settextcolor(7);
_outtext(xx1);

_settextwindow(top+15, left+17, bot+15, right+17);
sprintf(xx1, " %3.2f", ts);
_settextcolor(7);
_outtext(xx1);

_settextwindow(top+16, left+17, bot+16, right+17);
sprintf(xx1, " %3.3f", 10.0);
_settextcolor(7);
_outtext(xx1);          */

_settextwindow(top+19, left+16, bot+19, right+16);
sprintf(xx1, " %3.2f", 10.0);
_settextcolor(7);
_outtext(xx1);

_settextwindow(top+20, left+16, bot+20, right+16);
sprintf(xx1, " %2.1f", te);
_settextcolor(7);
_outtext(xx1);

_settextwindow(top+23, left+16, bot+23, right+16);
sprintf(xx1, " %3.2f", odssp);
_settextcolor(7);
_outtext(xx1);

_settextwindow(top+24, left+16, bot+24, right+16);
sprintf(xx1, " %3.2f", 10.0);
_settextcolor(7);
_outtext(xx1);

_settextwindow(top+26, left+17, bot+26, right+17);
sprintf(xx1, " %d", k);
_settextcolor(7);
_outtext(xx1);

```

```

}

int menui(int *TC,float *odssp)

{
    int i,sai;
    char sn;

    getch();
    _clearscreen(_GCLEARSCREEN);
    printf("\n  Deseja terminar e sair?      S/N\n");
    sai=0;
    sn=getch();
    if (sn=='s'){sai=1;}
    if (sn=='n'){

        printf("  Deseja esvaziar o coletor de po?  S/N\n");
        sn=getch();
        if (sn=='s'){
            /*massant= *mass;
            printf("      Pausa para a troca ( quando terminar tecle ENTER
)\n");
            getch();
            printf("\n");*/
        }
        if (sn=='n'){

            printf("  Mudar set-point de OD?  S/N\n");
            sn=getch();
            if (sn=='s'){
                printf("valor antigo      odssp=%f\n",*odssp);
                printf("novo valor      odssp=");scanf("%f",&(*odssp));
                printf("\n");
            }
            if (sn=='n'){

                /* printf("  Mudar set-point de Ys?  S/N\n");
                sn=getch();
                if (sn=='s'){
                    printf("valor antigo      Yssp=%f\n",*Yssp);
                    printf("novo valor      Yssp=");scanf("%f",&(*Yssp));
                    printf("\n");
                }
                if (sn=='n'){

                    */
            }
            if(sai==0){telagrafica();}
            return(sai);
        }
    }

void telagrafica(void)
{
    int i,j,top,left,bot,right;

    if (_setvideomode(_VRES16COLOR)==0)
        {printf("can't set mode");exit(1);}
    _setbkcolor(_RED);
    _setcolor(14);
    _rectangle(_GIFILLINTERIOR,435,0,639,479);
    _setcolor(4);
    _rectangle(_GIFILLINTERIOR,438,3,636,476);

```

```

_setcolor(14);
_rectangle(_GFillInterior, 438, 50, 636, 51);
_setcolor(14);
_rectangle(_GFillInterior, 0, 0, 636, 2);
_setcolor(14);
_rectangle(_GFillInterior, 0, 476, 636, 479);
_setcolor(14);
_rectangle(_GFillInterior, 0, 0, 3, 479);
top=2;
left=90;
bot=3;
right=57;
_settextwindow(top, left, bot, right);
_settextcolor(16);
_outtext("CONTROLADOR PREDITIVO");
_settextwindow(top+1, left, bot+1, right);
_settextcolor(14);
_outtext("          QDMC");

_settextwindow(top+3, left, bot+3, right);
_settextcolor(14);
_outtext("Variaveis Controladas");

_settextwindow(top+4, left, bot+4, right);
_settextcolor(2);
_outtext("ODs (mg/l) ");

_settextwindow(top+5, left, bot+5, right);
_settextcolor(2);
_outtext("pH ");

_settextwindow(top+7, left, bot+7, right);
_settextcolor(14);
_outtext("Variaveis Manipuladas");

_settextwindow(top+8, left, bot+8, right);
_settextcolor(2);
_outtext("rot. sop. (rpm) ");

_settextwindow(top+9, left, bot+9, right);
_settextcolor(2);
_outtext("ac/base W (g/s) ");

_settextwindow(top+11, left, bot+11, right);
_settextcolor(14);
_outtext("Variaveis Associadas");

_settextwindow(top+12, left, bot+12, right);
_settextcolor(2);
_outtext("v.ar Q (mc/min) ");

_settextwindow(top+13, left, bot+13, right);
_settextcolor(2);
_outtext("temp. Ts (oC) ");

/*_settextwindow(top+14, left, bot+14, right);
_settextcolor(2);
_outtext("temp. Te (oC) ");

_settextwindow(top+15, left, bot+15, right);
_settextcolor(2);

```

```

_outtext("temp. Ts (oC) ");

_settextwindow(top+16, left, bot+16, right);
_settextcolor(2);
_outtext("umid. FMs ");*/

_settextwindow(top+18, left, bot+18, right);
_settextcolor(14);
_outtext("Variaveis Perturbacao");

_settextwindow(top+19, left, bot+19, right);
_settextcolor(2);
_outtext("ODe (mg/l) ");

_settextwindow(top+20, left, bot+20, right);
_settextcolor(2);
_outtext("temp. Te (oC) ");

_settextwindow(top+22, left, bot+22, right);
_settextcolor(14);
_outtext("Valores Desejados");

_settextwindow(top+23, left, bot+23, right);
_settextcolor(2);
_outtext("s.p. ODs (mg/l) ");

_settextwindow(top+24, left, bot+24, right);
_settextcolor(2);
_outtext("s.p. pH ");

_settextwindow(top+26, left, bot+26, right);
_settextcolor(14);
_outtext("tempo (s) ");

_setcolor(3);
_rectangle(_GFillInterior, 45, 15, 215, 115);
_rectangle(_GFillInterior, 45, 130, 215, 230);
_rectangle(_GFillInterior, 45, 245, 215, 345);
_rectangle(_GFillInterior, 45, 360, 215, 460);

_rectangle(_GFillInterior, 250, 15, 420, 115);
_rectangle(_GFillInterior, 250, 130, 420, 230);
_rectangle(_GFillInterior, 250, 245, 420, 345);
_rectangle(_GFillInterior, 250, 360, 420, 460);
for (i=1; i<21; i++){
    j=i*5;
    _setcolor(1);
    if(i==5){_setcolor(0);}
    if(i==10){_setcolor(0);}
    if(i==15){_setcolor(0);}

    _moveto(45, 15+j);
    _lineto(215, 15+j);
    _moveto(45, 130+j);
    _lineto(215, 130+j);
    _moveto(45, 245+j);
    _lineto(215, 245+j);
    _moveto(45, 360+j);
    _lineto(215, 360+j);

    _moveto(250, 15+j);

```

```

    _lineto(420,15+j);
    _moveto(250,130+j);
    _lineto(420,130+j);
    _moveto(250,245+j);
    _lineto(420,245+j);
    _moveto(250,360+j);
    _lineto(420,360+j);

}
for (i=1;i<35;i++){
    j=i*5;
    _moveto(45+j,15);
    _lineto(45+j,115);
    _moveto(45+j,130);
    _lineto(45+j,230);
    _moveto(45+j,245);
    _lineto(45+j,345);
    _moveto(45+j,360);
    _lineto(45+j,460);

    _moveto(250+j,15);
    _lineto(250+j,115);
    _moveto(250+j,130);
    _lineto(250+j,230);
    _moveto(250+j,245);
    _lineto(250+j,345);
    _moveto(250+j,360);
    _lineto(250+j,460);

}
_setcolor(14);
_settextwindow(2,3,35,9);
_settextcolor(11);
_outtext("    15\n\nODs\n        \n\n\n    0\n    50\n\nTs\nnoC\n\n\n\n
0\n3.5\n\nnrot\nnrpm\n\n\n\n    0\n 14\n\n\nnpH\n\n\n\n    0");
_settextwindow(2,29,34,35);
_outtext("    15\n\nODE\n\n\n\n\n    0\n    50\n\nTe\nnoC\n\n\n\n    0\n0.05\n\n
Q\nmc/\ns    \n\n    0\n0.1\n\n W\ng/s\n\n\n\n    0");
}

```