

Centro Federal de Educação Tecnológica de Minas Gerais
Campus Divinópolis
Graduação em Engenharia Mecatrônica

Álan Crístopher e Sousa

IMPLEMENTAÇÃO DE CONTROLADOR MPC EM UM
SISTEMA A PARÂMETROS DISTRIBUÍDOS VIA SUBSISTEMAS
INTERCONECTADOS



Divinópolis
2018

Álan Crístopher e Sousa

IMPLEMENTAÇÃO DE CONTROLADOR MPC EM UM
SISTEMA A PARÂMETROS DISTRIBUÍDOS VIA SUBSISTEMAS
INTERCONECTADOS

Monografia de Trabalho de Conclusão de Curso
apresentada ao Colegiado de Graduação em Enge-
nharia Mecatrônica como parte dos requisitos exi-
gidos para a obtenção do título de Engenheiro Me-
catrônico.

Eixo de Formação: Modelagem e Controle de Pro-
cessos.

Orientador: Prof. Dr. Valter Júnior de Souza Leite

Co-orientador: Prof. Dr. Ignacio Rubio Scola



Divinópolis

2018

Resumo

Controle preditivo baseado em modelo (MPC — *model predictive control*) é uma técnica avançada de controle que permite a inserção de restrições no equacionamento do controlador. Seu objetivo é encontrar a trajetória de controle ótima que respeite as restrições impostas, que podem ser na amplitude ou variação do sinal de controle, de saída ou de um estado. Ele já está estabelecida em indústrias que lidam com processos multi-variáveis de dinâmica lenta, especialmente na indústria petro-química. Sua formulação mais comum é utilizando modelos por espaço de estados, o que requer que todos os estados sejam conhecidos. Como nem sempre é possível medi-los, faz-se necessário o uso de observadores, que são técnicas de estimar os estados do sistema a partir do modelo, sinal de controle e um estado medido. Utilizando técnicas de modelagem pouco exploradas, como sistemas a parâmetros distribuídos (SPD), pode-se utilizar observadores para estimar parâmetros do sistema em pontos de interesse onde não é possível ou viável inserir sensores. Assim pode-se, por exemplo, medir-se a temperatura na extremidade de um sólido e, através desta, recuperar-se a temperatura em algum ponto no meio do sólido. Ao combinar as três técnicas é possível controlar uma variável em um ponto diferente daquele sendo medido. Assim propõe-se o desenvolvimento e implementação de um controlador MPC que utilize SPD para realizar o controle de uma variável estimada em um ponto intermediário. Para isso será utilizada a planta presente no laboratório de sinais e sistemas. O modelo SPD utilizado será o desenvolvido por Barroso (2017). A implementação será feita utilizando os softwares *Moirai* e *Lachesis*, desenvolvidos *in loco*, que serão atualizados de forma a comunicar com o *hardware* da planta e receberá atualizações visando sua melhoria. Também será feita alteração na microeletrônica da planta, com a finalidade de prover um meio alternativo para acionamento do forno. Pretende-se com este trabalho, que envolve principalmente as áreas de controle e computação, aprofundar os estudos do grupo de modelagem e controle de processos mecatrônicos em SPD e MPC, bem como facilitar futuros trabalhos nesta e outras plantas, através do melhor desenvolvimento e teste da plataforma de controle.

Palavras-chave: Controle preditivo por modelo, sistema a parâmetros distribuídos, observador de Kalman

Abstract

Model predictive control (MPC) is an advanced control technique that allows the insertion of constraints in the controller equation. Its objective is to find the optimal control trajectory that respects the constraints imposed, which can be in the amplitude or variation of the control signal, output or a state. It is already established in industries dealing with multi-variable processes of slow dynamics, especially in the petrochemical industry. Its most common formulation is to use state-space models, which requires all states to be known. As it is not always possible to measure them, it is necessary to use observers, which are techniques of estimating the states of the system from the model, control signal and a measured state. Using less exploited modeling techniques, such as distributed parameters systems (DPS), observers can be used to estimate system parameters at points of interest where it is not possible or feasible to insert sensors. Thus, for example, one can measure the temperature at the end of a solid and therethrough recover the temperature at some point in the middle of the solid. By combining the three techniques it is possible to control a variable at a point other than the one being measured. Thus it is proposed the development and implementation of an MPC controller that uses SPD to perform the control of an estimated variable in an intermediate point. For this, the plant will be used in the signal and system laboratory. The SPD model used will be the one developed by Barroso (2017). The implementation will be made using the Moirai and Lachesis softwares, developed *in loco*, which will be updated in order to communicate with the plant hardware and receive updates to improve it. Also, there will be made change in the microelectronics of the plant, in order to provide an alternative means to drive it. It is intended that this work, which mainly involves the areas of control and computation, to deepen the studies of the modeling and control group of mechatronic processes in DPS and MPC, as well as to facilitate future works in this and other plants, through the best development and test of the control platform.

Keywords: Model predictive control, distributed parameters system, Kalman filter

Sumário

| | | |
|----------|--|-----------|
| 1 | Introdução | 1 |
| 1.1 | Definição do Problema | 2 |
| 1.2 | Objetivos | 3 |
| 1.3 | Motivação | 4 |
| 1.4 | Estado da arte | 4 |
| 2 | Revisão da literatura | 7 |
| 3 | Metodologia | 9 |
| 4 | Fundamentação Teórica | 11 |
| 4.1 | Controlador preditivo por modelo | 11 |
| 4.1.1 | Princípio de funcionamento | 11 |
| 4.1.2 | Restrições de parâmetros | 15 |
| 4.2 | Sistemas a parâmetros distribuídos | 18 |
| 4.2.1 | Modelagem do forno por LPVs interconectados | 20 |
| 4.3 | Observadores | 22 |
| 4.3.1 | Filtro de Kalman | 23 |
| 4.3.2 | Observador exponencial com fator de esquecimento | 24 |
| 4.3.3 | Observabilidade | 24 |
| 5 | Resultados e Discussões | 25 |
| 5.1 | Estudos teóricos | 25 |
| 5.1.1 | Controle preditivo por modelo | 25 |
| 5.1.2 | Modelagem térmica e SPD | 27 |

| | | |
|----------|---|-----------|
| 5.2 | Modificação do hardware e implantação da plataforma | 28 |
| 5.3 | Calibrações, testes e certificações | 28 |
| 5.4 | Melhorias na plataforma | 29 |
| 6 | Conclusão | 33 |
| | Bibliografia | 35 |

Lista de ilustrações

| | |
|--|----|
| Figura 1 – Sensores da planta | 3 |
| Figura 2 – Subsistemas interconectados | 18 |
| Figura 3 – Subsistema LPV | 19 |
| Figura 4 – Subsistemas LPV interconectados | 20 |
| Figura 5 – Sistema de tanques comunicantes | 26 |
| Figura 6 – Simulação do modelo dos tanques com controlador MPC | 26 |
| Figura 7 – MPC aplicado ao sistema real | 27 |
| Figura 8 – SPD simulado | 27 |
| Figura 9 – SPD simulado com observador exponencial | 28 |

Lista de tabelas

| | |
|---|----|
| Tabela 3 – Coeficientes da calibração | 29 |
|---|----|

Lista de acrônimos e notações

| | |
|-------|---|
| ARMAX | Modelo autoregressivo de média variável com entradas exógenas, do inglês <i>Autoregressive–moving-average with exogenous inputs model</i> |
| DMC | Controle Dinâmico por Matriz, do inglês <i>Dynamic Matrix Control</i> |
| EDP | Equação Diferencial Parcial |
| IMC | Controle por Modelo Interno, do inglês <i>Internal Model Control</i> |
| LPV | Variação linear de parâmetros, do inglês <i>Linear Parameter Varying</i> |
| MPC | Controle preditivo por modelo, do inglês <i>Model Predictive Control</i> |
| SPC | Sistema a Parâmetros Concentrados |
| SPD | Sistema a Parâmetros Distribuídos |

Introdução

Controle preditivo por modelo (MPC — *Model Predictive Control*) é uma técnica de controle avançada que utiliza o modelo do sistema para prever a saída em momentos futuros e, com isso, gerar uma estratégia de controle. Esta técnica pode ser aplicada em modelos contínuos, onde o tempo de amostragem é tão pequeno que pode ser desconsiderado, ou discretos, onde o modelo já espera que exista um certo tempo entre amostragens. A abordagem discreta será usada pois melhor ilustra o funcionamento do MPC, além de fornecer ao controlador mais tempo para o processamento do próximo sinal de controle.

Em um modelo discreto o sinal é amostrado a cada Δt segundos. O modelo não varia mais diretamente com o tempo, mas sim com o instante de amostragem k , que se associa com o tempo através do tempo de amostragem como $t = k\Delta t$. Assim, em um modelo discreto, ao se passar do tempo 0 para o tempo Δt ou do tempo Δt para o tempo $2\Delta t$, dizemos que se passou um instante.

Utilizando um modelo discreto pode-se então prever a saída do sistema para um instante futuro, se as entradas que serão aplicadas ao sistema nos instantes anteriores forem conhecidas. Pode-se então prever as saídas dos próximos N_p instantes e utilizar técnicas de otimização para obtermos os próximos N_c sinais de controle ótimos que levarão o sistema à referência. Ao utilizar apenas o primeiro sinal de controle e recalculá-lo os sinais de controle ótimos a todo instante, o controlador será capaz de rejeitar ruídos e reagir a mudanças, como a da referência. Esta técnica é chamada de Controle por Horizonte Recessivo e é a base do MPC (WANG, 2009).

A discretização foi feita no tempo, pois altera-se o modelo para que este não mais dependa do tempo mas sim de um instante. Sistemas que dependem apenas do tempo

são chamados de sistemas a parâmetros concentrados (SPC). No entanto existem sistemas que não variam apenas no tempo, mas também no espaço, por exemplo. Um exemplo é a transferência de calor em uma barra metálica, onde a temperatura depende tanto do tempo quanto do ponto no espaço em que se faz a medição. Sistemas que dependem de mais de uma variável são chamados de sistemas a parâmetros distribuídos (SPD). Sua modelagem é feita utilizando derivadas parciais e equações diferenciais parciais (EDP) (BARROSO, 2017).

Para ambos tipos de modelos é comum usar a modelagem por espaço de estados ao se trabalhar com MPC. Isso permite controlar facilmente sistemas com múltiplas entradas e múltiplas saídas. No entanto, ao se trabalhar com espaço de estados, é necessário que o valor de todos os estados sejam conhecidos, o que nem sempre é possível, seja por limitações do sistema ou até mesmo físicas. Para resolver este problema utiliza-se observadores.

O observador é um sistema que estima os estados de um modelo baseado nas entradas e saídas do mesmo. O observador de Kalman, também conhecido como filtro de Kalman, é uma formulação que permite estimar o modelo de forma precisa mesmo na presença de ruídos, seja na leitura da saída ou dos próprios estados (WANG, 2009).

1.1 Definição do Problema

A Figura 1 mostra os sensores do forno presente no Laboratório de Sinais e Sistemas. Nela podemos ver os sensores S_1 a S_5 e o fluxo de ar quente q . Imagine que queiramos controlar a temperatura na posição onde encontra-se o sensor S_3 , mas que apenas o sensor S_5 esteja funcionando. Com um modelo SPD podemos ter dois modelos SPC: um que modele o fluxo de temperatura até o sensor S_3 e um outro que modele de S_3 até S_5 , ambos interconectados. Desta forma podemos utilizar um observador e a leitura do sensor S_5 para estimar a temperatura no sensor S_3 , e controlá-la mesmo sem fazer sua medição direta.

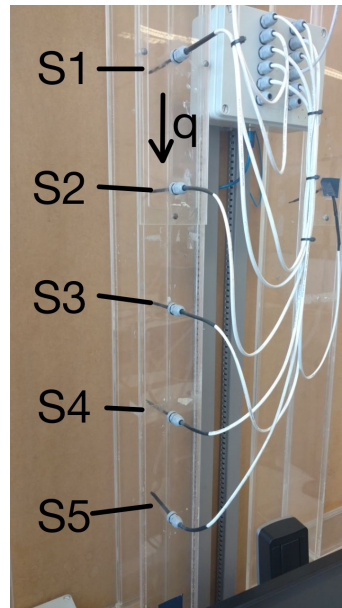


Figura 1 – Sensores da planta

1.2 Objetivos

O objetivo desse trabalho é desenvolver um controlador do tipo MPC com restrições com modelo SPD utilizando observadores do tipo Kalman e implementá-lo utilizando e adaptando a plataforma de controle desenvolvida pelo proponente para uso no forno do laboratório de sinais e sistemas. O objetivo pode ser assim dividido:

- modificar a eletrônica da planta: instalar circuitos microprocessados com o intuito de controlar o acionamento do forno, fornecendo um caminho alternativo ao acionamento que já está implementado e permitindo a integração com a plataforma de controle;
- modificar a plataforma de controle: desenvolver driver específico para o forno, de forma a tornar o acionamento dos atuadores e a leitura dos sensores mais simples e direto para os futuros usuários;
- desenvolver o controlador MPC: utilizar o modelo SPD desenvolvido pelo Barroso (2017) para desenvolver um controlador MPC com restrições na entrada e saída de forma a controlar a temperatura em um ponto diferente daquele onde está fisicamente instalado o sensor;
- implementar o controlador: utilizar a linguagem Python e a plataforma de controle para implementar o controlador e executar o controle da planta;

- comparar o desempenho do controlador: usar índices de desempenho para comparar o desempenho do controlador MPC com controladores PI ou PID a serem desenvolvidos utilizando as técnicas de Clarke (1984) e Martins (2005);
- realizar melhorias na plataforma: ao utilizar a plataforma como usuário, espera-se encontrar dificuldades, erros e novas ideias que serão ser corrigidos/implementadas, como, por exemplo, recuperação de mensagens de erro salvas no banco de dados e exibição para o usuário e verificação de sintaxe de código digitado pelo usuário.

1.3 Motivação

A principal motivação para o trabalho é poder fazer o controle de um processo sem medir diretamente a variável controlada. Isto é interessante pois nem sempre é possível medir diretamente a grandeza que se deseja controlar, seja por restrições físicas (como, por exemplo, colocar um sensor no centro de um alto-forno) ou por questões financeiras.

A escolha do controlador MPC se tornou interessante por ser uma técnica avançada de controle que ainda não é estudada no *campus V*. Assim, além da possibilidade de estudar uma técnica avançada, este trabalho contribui por ser o primeiro trabalho a utilizar tal controlador no Grupo de Modelagem e Controle de Sistemas Mecatrônicos.

1.4 Estado da arte

O uso de controladores MPC com modelos SPD não é novo. Sua maior aplicação, no entanto, continua sendo na área da química, fabricação de aço e, principalmente na indústria petroquímica. Ele foi desenvolvido inicialmente para controle de processos químicos, onde o equipamento é caro, o processo é lento e contém várias entradas, saídas e estados que devem ser restringidas em amplitude e taxa de variação (CAIRANO, 2012).

Este cenário está mudando e outras indústrias — com modelos com poucas entradas e saídas, custo mais baixo e dinâmica mais rápida — estão adotando os controladores MPC. No entanto, isso apresenta alguns desafios no desenvolvimento de controladores, pois o *framework* MPC não foi desenvolvido para tais sistemas e possui limitações como, por exemplo, o alto custo computacional. Trabalhos já estão sendo desenvolvidos para minimizar ou sanir tais problemas (CAIRANO, 2012).

Em ambientes industriais é comum o uso de computadores lógicos programáveis para fazer a interface com o hardware. No meio acadêmico, no entanto, utiliza-se softwares como o *MATLAB* e *LabVIEW* para a prototipagem rápida. Isto requer que o pesquisador recrie a interface gráfica e controle a aquisição de dados manualmente, o que consome tempo que poderia ser gasto com a pesquisa.

Assim, uma plataforma que faça a interface com o hardware e possibilite a execução de um controlador de forma transparente permite que o pesquisador se concentre nestes estudos e não se preocupe com detalhes da implementação da aquisição dos dados. Tal ferramenta também permitiria ao pesquisador interagir com CLPs, desenvolvendo e testando controladores de forma fácil em uma linguagem simples (Python) sem se preocupar com todas as nuances das linguagens LADDER utilizadas nos CLPs, podendo implementar seu controlador nesta linguagem limitada apenas ao final, quando este tiver sido devidamente testado e otimizado.

A aplicação de uma plataforma para controle em todas as plantas do laboratório permite que os usuários possam migrar de uma planta para outra sem dificuldades. Também permite que controladores e observadores desenvolvidos sejam testados em diferentes plantas com poucas modificações, já que as interfaces seriam as mesmas em todas elas.

Revisão da literatura

A malha fechada foi formalizada matematicamente em 1868 por Maxwell. Em 1927 Black (1934) demonstrou a utilidade da realimentação negativa nos laboratórios da Bell onde aplicou a malha fechada com realimentação negativa nos amplificadores das linhas de transmissão.

No entanto, ao ampliar o sinal, foi ampliado também o ruído. Para resolver este problema iniciaram-se estudos para utilizar técnicas descritas por Laplace, Fourier e Cauchy, que propunham o uso do domínio da frequência para modelar sistemas dinâmicos. O problema passou a ser filtrar as frequências desejadas de forma a não aplicar o ganho da malha fechada no ruído (BRYSON; HO; SIOURIS, 1979).

Nyquist e Bode (1940) criaram ferramentas na década de 1930 que permitem quantizar a estabilidade de um sistema em malha fechada. O uso de tais ferramentas facilitou o desenvolvimento de controladores que alteram a dinâmica do sistema para uma dinâmica desejada. A adição de ferramentas estocásticas permitiu também o desenvolvimento de filtros ótimos.

Com estas técnicas estabeleceu-se o que chamamos de controle clássico. Este é caracterizado pelo desenvolvimento no domínio da frequência e pelo uso de técnicas projetadas para que seus cálculos fossem feitos à mão, ou no máximo com o uso de tabelas (DORF; BISHOP, 2010).

Embora o uso da análise no domínio da frequência tenha sido útil principalmente para sua época, começaram a aparecer sistemas mais complexos, com várias entradas e saídas, ordem elevada e não lineares, que não eram bem descritos no domínio da frequência. Nesta época, estudos no domínio do tempo, utilizando as equações diferenciais, começaram a

aparecer, principalmente para resolver problemas da indústria aeroespacial (LYAPUNOV, 1892).

Várias técnicas foram desenvolvidas nesta época. Uma se destaca: DMC — *Dynamic Matrix Control*. Ela visa resolver os problemas de controle multivariável com restrições, típico nas indústrias química e petroquímica. Antes do desenvolvimento desta técnica o controle era feito por várias malhas em cascata (CUTLER; RAMAKER, 1980).

Seu impacto na indústria foi gigante. Provavelmente não há nenhuma empresa extratora de petróleo que não utilize esta técnica ou uma derivada. O desenvolvimento inicial do MPC se deu como uma tentativa de entender o DMC, que parecia desafiar a análise teórica tradicional por ser formulado de maneira não convencional. Por exemplo, outra técnica, a IMC — *Internal Model Control* — falhou em explicar o funcionamento da DMC mas acabou ajudando no desenvolvimento do controle robusto (MORARI; LEE, 1999).

Hoje em dia o controle preditivo é formulado sempre em espaço de estados, mesmo que seja possível sua formulação por, por exemplo, funções de transferência. O controle preditivo por modelo é uma das formas de controle desenvolvidas com a ideia de buscar a trajetória ótima de controle, que observa-se sempre existir em sistemas dinâmicos. Para isso é utilizada a otimização custo, normalmente uma função de energia (MORARI; LEE, 1999; BRYSON; HO; SIOURIS, 1979).

O grande problema destes controladores hoje é o custo computacional quando se necessita de um controlador *on-line*, o que normalmente é o que se deseja, pois estes retornam resultados melhores na presença de restrições. Os estudos atuais seguem duas linhas: melhorar os *solvers* de forma a melhorar o desempenho ou encontrar maneiras de pré-processar as restrições, para que apenas multiplicações matriciais sejam realizadas em tempo real. A primeira linha de pesquisa é a mais explorada hoje em dia (WANG, 2009; ZHANG, 2016).

Metodologia

Para que a plataforma de controle possa se comunicar com o forno do laboratório de sinais e sistemas será necessário modificar o circuito de acionamento da mesma. Para isso serão utilizados circuitos microcontrolados que irão se comunicar com o circuito de acionamento e sensoramento presente na mesma. Um estudo da forma como isso é feito hoje será realizado com a finalidade de inserir um circuito paralelo ao atual, de forma que o usuário possa escolher qual circuito de acionamento e aquisição de dados será utilizado.

Esse novo circuito será programado para se comunicar com a plataforma de controle. Um *driver* será escrito para a mesma para permitir o controle dos atuadores e leitura dos sensores de forma transparente e específica para este sistema. O *driver* irá abstrair a interface com o hardware de forma que o usuário não tenha que se preocupar com detalhes da implementação, como número de porta e pino, tensões de operação, etc.

Para que se possa trabalhar com a planta também será necessário calibrar todos os sensores e atuadores, além de se fazer uma conferência dos circuitos elétricos, para certificar que estes ainda funcionam de forma correta. Todo este procedimento será feito usando-se a plataforma de controle para a aquisição de dados. Para isso será necessário estudar os trabalhos de Barroso (2015) e Silva (2014). Como prova de conceito, também será utilizado um computador portátil de baixo custo para o controle da planta.

Os modelos levantados por Barroso (2017) serão validados no sistema físico. Isto é necessário para verificar que não houve alterações significativas no sistema desde o seu desenvolvimento. Para o uso destes modelos será necessário estudar sobre sistemas a parâmetros distribuídos e observadores de Kalman.

O controlador MPC será discreto no tempo e baseado em modelo em espaço de estados,

logo será necessário estudar sobre controle digital e modelagem em espaço de estados, além da própria técnica de controle. Para isto serão utilizados, principalmente, os livros do Dorf e Bishop (2010), do Liberty (1972), do Wang (2009) e as notas de aula do Professor Patwardhan (2014).

Para o desenvolvimento do controlador será utilizado o modelo SPD e serão determinadas restrições do sinal de controle. Os estados utilizados no controlador serão os provenientes do observador do tipo Kalman utilizado pelo Barroso (2017).

Para comparar o desempenho do controlador MPC serão desenvolvidos controladores PI utilizando as técnicas descritas por Clarke (1984) e Martins (2005). Serão utilizados os índices IAE e IVx para realizar uma comparação quantitativa dos controladores.

Todos os controladores serão implementados utilizando a plataforma de controle e a linguagem de programação Python, assim como todos os testes que requeiram acionamento do forno e/ou medição de seus termômetros. Desta forma serão corrigidos na plataforma todos os problemas percebidos na mesma do ponto de vista de usuário. Também serão implementadas as funcionalidades que se mostrem necessárias e/ou desejadas, visando seu aperfeiçoamento.

Fundamentação Teórica

Para a realização deste trabalho é necessário conhecimento nas áreas de controle preditivo por modelo e sistemas a parâmetros distribuídos. Este capítulo visa explicar sucintamente esses conceitos.

4.1 Controlador preditivo por modelo

Controle preditivo por modelo é uma técnica comumente utilizada pelas indústrias petroquímica, farmacêutica e do aço. Ela utiliza o modelo para prever a saída do sistema e, com isso, otimizar a trajetória do sinal de controle (WANG, 2009).

4.1.1 Princípio de funcionamento

A formulação MPC mais comum é a em espaço de estados. Um modelo em espaço de estados tem a forma

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t),\end{aligned}\tag{4.1}$$

onde $A \in R^{n \times n}$ é a matriz dinâmica do sistema, $B \in R^{n \times m}$ e a matriz de entrada, $C \in R^{m \times n}$ é a matriz de saída. $D \in R^{m \times m}$ é normalmente 0. $x(t) \in R^{n \times 1}$ é o vetor de estados e $u(t) \in R^{m \times 1}$ são as entradas do sistema no instante t .

Neste trabalho será utilizada a formulação discreta do controlador preditivo. Esta formulação tem a vantagem de poder ser implementada facilmente em dispositivos digitais, além de permitir certo controle do tempo de amostragem.

Ao discretizar um sistema altera-se sua formulação para que a matriz de dinâmica do sistema não mais produza a variação do estado, mas sim o valor do estado após Δt segundos. Um intervalo de tempo de Δt segundos é chamado de instante, sendo $t = k\Delta t$. Para discretizar um modelo em espaço de estados usa-se a equação (4.2).

$$x(k+1) = e^{AT}x(k) + \int_0^T e^{At}dtBu(k) \quad (4.2)$$

Ao discretizar o modelo (4.1) utilizando (4.2) obtém-se (4.3).

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ y(k) &= Cx(k) + Du(k). \end{aligned} \quad (4.3)$$

Uma vez obtido o modelo discreto o mesmo deve ser aumentado com um integrador. Para inserir um integrador no sistema utiliza-se a equação (4.4).

$$\begin{bmatrix} \Delta x(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} A & 0_m^T \\ C \cdot A & 1 \end{bmatrix} \begin{bmatrix} \Delta x(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B \\ C \cdot B \end{bmatrix} \Delta u(k) \quad (4.4)$$

Observe que os estados e o sinal de entrada foram substituídos por variações dos mesmos. Isso é possível pois

$$\Delta x(k+1) = A\Delta x(k) + B\Delta u(k), \quad (4.5)$$

e é também necessário para o funcionamento do controlador. O motivo para esta reformulação será percebido mais adiante.

Uma vantagem do modelo em espaço de estados discreto é que é possível calcular estados futuros facilmente, sem a necessidade de se resolver uma equação diferencial. Utilizando deste fato pode-se utilizar o modelo (4.3) para obter uma equação da próxima saída do sistema, através da substituição de $x(k)$ por $x(k+1)$ na equação da saída:

$$y(k+1) = C(Ax(k) + Bu(k)). \quad (4.6)$$

Ao realizar esta substituição uma segunda vez, obtém-se a saída do sistema após 2 instantes:

$$y(k+2) = C(A(Ax(k) + Bu(k)) + Bu(k+1)). \quad (4.7)$$

Com uma nova substituição obtém-se a saída após 3 instantes:

$$y(k+3) = C(A(A(Ax(k) + Bu(k)) + Bu(k+1)) + Bu(k+2)). \quad (4.8)$$

Tem-se então que a saída depende do estado atual e da trajetória de controle. Reescrevendo estas equações de forma matricial para qualquer número N_p de saídas futuras e N_c de entradas futuras, tem-se

$$Y = Fx(k) + \Phi U \quad (4.9)$$

onde

$$Y = \begin{bmatrix} y(k+1) \\ y(k+2) \\ y(k+3) \\ \vdots \\ y(k+N_p) \end{bmatrix}; F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}; U = \begin{bmatrix} u(k+1) \\ u(k+2) \\ u(k+3) \\ \vdots \\ u(k+N_c) \end{bmatrix}; \quad (4.10)$$

$$\Phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0 \\ CAB & CB & 0 & \dots & 0 \\ CA^2B & CAB & CB & \dots & 0 \\ \vdots & & & & \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \dots & CA^{N_p-N_c}B \end{bmatrix}$$

No entanto, para que essa formulação funcione devemos ter $N_c = N_p$. Isto no entanto não é desejado. Por questões de implementação computacional e matemáticas é preferível que $N_c \ll N_p$. Isso se torna possível se substituirmos U por ΔU , o que é feito na matriz aumentada. Essa substituição é possível, pois, devido à presença do integrador, um valor de $\Delta U = 0$ significa que o último sinal aplicado é mantido.

Para compreender melhor este conceito tome com exemplo um sistema de tanques onde deseja-se controlar o nível alterando-se a vazão de uma bomba. Para todo $N_p > N_c$ tem-se uma vazão de 0, ou seja, a bomba seria desligada. Isso faria com que o nível caísse toda vez que o sistema alcançasse o equilíbrio. Utilizando ΔU o valor zero significa manter a última vazão aplicada, o que faz com que o sistema permaneça em equilíbrio uma vez que o sinal de controle seja zerado.

MPC é uma técnica baseada em otimização. Problemas de otimização são resolvidos encontrando-se o máximo ou mínimo de alguma função. Deve-se então definir uma função que, ao ser minimizada, terá como resultado a trajetória ótima. Este tipo de função é chamada de função de custo e normalmente utiliza-se em MPC uma função quadrática, o que garante a convergência. Escrevendo a função com o intuito de minimizar o erro em regime permanente e a variação do sinal de controle, tem-se

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U, \quad (4.11)$$

onde R_s é o vetor referência com a mesma dimensão do vetor Y e \bar{R} é uma variável que pode ser usada para controlar o peso da variação do sinal de controle. \bar{R} maior implica em variações menores. Substituindo (4.9) em (4.11) tem-se

$$J = (R_s - Fx(k))^T (R_s - Fx(k)) - 2\Delta U^T \Phi^T (R_s - Fx(k)) + \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U. \quad (4.12)$$

Derivando em função da variação do sinal de controle, ΔU :

$$\frac{\partial J}{\partial \Delta U} = -2\Phi^T (R_s - Fx(k)) + 2(\Phi^T \Phi + \bar{R}) \Delta U. \quad (4.13)$$

Igualando $\frac{\partial J}{\partial \Delta U}$ a zero e isolando ΔU :

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - Fx(k)). \quad (4.14)$$

Ao utilizar $N_c = N_p$, é possível, através da equação (4.14), encontrar os valores ótimos de ΔU que farão o sistema seguir uma referência. No entanto, o controlador não será capaz de rejeitar ruídos. Aplicando apenas o primeiro sinal de controle, medindo ou estimando os estados do sistema e recalculando a trajetória a cada instante é possível contornar este problema.

Esta é a formulação básica do MPC, porém ela não fornece o que pode ser considerada a maior vantagem desta técnica de controle: restrição do sinal de controle, da saída, de estados ou de suas variações. Para isto é necessário voltar na função de custo e resolvê-la de uma maneira que permita a inserção de restrições no equacionamento.

4.1.2 Restrições de parâmetros

Os parâmetros mais comumente restringidos são a saída do sistema, o sinal de controle e a variação do sinal de controle. Estas restrições tem a forma de inequações e podem ser escritas como

$$U^{\min} \leq U \leq U^{\max}, \quad (4.15)$$

$$Y^{\min} \leq Y \leq Y^{\max}, \quad (4.16)$$

e

$$\Delta U^{\min} \leq \Delta U \leq \Delta U^{\max}. \quad (4.17)$$

Usando a equação (4.17) como exemplo, pode-se reescrevê-la como duas inequações:

$$\begin{aligned} -\Delta U &\leq -\Delta U^{\min} \\ \Delta U &\leq \Delta U^{\max}. \end{aligned} \quad (4.18)$$

Em forma matricial,

$$\begin{bmatrix} -I \\ I \end{bmatrix} \Delta U \leq \begin{bmatrix} -\Delta U^{\min} \\ \Delta U^{\max} \end{bmatrix}. \quad (4.19)$$

Para os casos de Y e U , deve-se fazer as devidas substituições de forma que a variável ΔU apareça na inequação. Esta substituição é necessária pois a técnica de otimização irá buscar a solução ótima para apenas uma variável (ΔU), sendo necessário que todas as restrições sejam funções desta variável.

Para Y deve-se substituir a equação (4.9) em (4.16), isolar $\Phi \Delta U$ e reescrevê-la de forma matricial:

$$\begin{bmatrix} -\Phi \\ \Phi \end{bmatrix} \Delta U \leq \begin{bmatrix} -Y^{\min} + Fx(k) \\ Y^{\max} - Fx(k) \end{bmatrix}. \quad (4.20)$$

Para a restrição de U , lembrando que $u(k) = u(k-1) + \Delta u(k)$, tem-se

$$\begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_c-1) \end{bmatrix} = \begin{bmatrix} I \\ I \\ \vdots \\ I \end{bmatrix} u(k-1) + \begin{bmatrix} I & 0 & \dots & 0 \\ I & I & \dots & 0 \\ \vdots & & & \\ I & I & \dots & I \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_c-1) \end{bmatrix}, \quad (4.21)$$

que pode ser reescrita como

$$\begin{bmatrix} -C_1 u(k-1) - C_2 \Delta U \\ C_1 u(k-1) + C_2 \Delta U \end{bmatrix} \leq \begin{bmatrix} -U^{\min} \\ U^{\max} \end{bmatrix}. \quad (4.22)$$

Reduzindo estas restrições em uma única inequação matricial, obtém-se

$$\begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} \Delta U \leq \begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix}, \quad (4.23)$$

sendo

$$\begin{aligned} M_1 &= \begin{bmatrix} -C_2 \\ C_2 \end{bmatrix}; N_1 = \begin{bmatrix} -U^{\min} + C_1 u(k-1) \\ U^{\max} + C_1 u(k-1) \end{bmatrix}; M_2 = \begin{bmatrix} -I \\ I \end{bmatrix}; \\ N_2 &= \begin{bmatrix} -\Delta U^{\min} \\ \Delta U^{\max} \end{bmatrix}; M_3 = \begin{bmatrix} -\Phi \\ \Phi \end{bmatrix}; N_3 = \begin{bmatrix} -Y^{\min} + Fx(k) \\ Y^{\max} - Fx(k) \end{bmatrix}. \end{aligned}$$

Pode-se inserir outras restrições, como em estados, por exemplo, seguindo essa lógica de formulação. Obtem-se então todas as restrições descritas na forma

$$M \Delta U \leq \gamma. \quad (4.24)$$

Este formato de restrições é comumente utilizado em problemas de otimização. Os métodos de programação quadrática, multiplicadores de lagrange e até mesmo o método SIMPLEX usam restrições escritas nesse formato. Deve-se então escolher um método de otimização e ajustar a função de custo à aquela necessária para usar o método. Como a função de custo é quadrática, convém escolher um método que seja capaz de otimizar problemas quadráticos.

O método de programação quadrática tem uma função de custo que se assemelha à que foi definida, como pode ser visto na equação (4.25). Vale notar que as variáveis e constantes desta função não tem relação com as variáveis definidas anteriormente, mas são as comumente utilizadas por matemáticos ao trabalhar com estes métodos.

$$J = \frac{1}{2}x^T E x + x^T F. \quad (4.25)$$

Reescrevendo (4.11) no formato de (4.25), encontra-se

$$\begin{aligned} E &= \Phi^T \Phi + \bar{R} \\ F &= \Phi^T (F x - R_s), \end{aligned} \quad (4.26)$$

no qual E e F à esquerda da igualdade vem da equação (4.25) e Φ , F , R_s e \bar{R} à direita da igualdade da formulação do MPC.

Explicar o funcionamento do método de programação quadrática foge do escopo deste trabalho. Existem bibliotecas que implementam este método em MATLAB e Python como, por exemplo, a CVXOPT¹. Para usar o *solver* de programação quadrática definido nessas bibliotecas deve-se definir as matrizes E , F , M e γ .

Analisando a equação (4.23) pode-se ver que suas dimensões não são iguais, isto é, a matriz M não é quadrada. Isso ocorre pois é possível ter mais restrições do que variáveis. Isso não é um problema já que as restrições são inequações, logo uma mais de uma restrição pode ser satisfeita se tentar satisfazer apenas uma delas, mais restritiva.

Em outras palavra, apenas um subconjunto de restrições estará ativo em um dado momento. Existem técnicas para determinar qual é o conjunto ativo. Isto, no entanto, não é necessário, já que o *solver* irá encontrar a mesma solução para o conjunto completo que para o subconjunto ativo de restrições. A vantagem de encontrar o conjunto ativo é reduzir o custo computacional da otimização.

¹ <https://cvxopt.org/>

4.2 Sistemas a parâmetros distribuídos

Neste trabalho é estudada apenas a abordagem proposta por Barroso (2017). Ela consiste basicamente em unir duas propostas já existentes, a SPD por subsistemas interconectados e a baseada em modelos lineares a parâmetros variantes no espaço (Spatial LPV, do inglês *Spatial Linear Parameter Varying*).

SPD por subsistemas interconectados consiste em modelar um problema que varia no espaço, ou seja, um SPD, utilizando vários sistemas a parâmetros concentrados. Assim cada SPC representa a saída do sistema em um ponto no espaço. Esta abordagem, no entanto, é limitada, já que os pontos no espaço onde pode-se fazer a estimação dos estados é fixada pelo modelo. Para fazer a estimação em outros pontos é necessário remodelar o sistema.

Na Figura 2 pode-se ver um esquemático do funcionamento desta aproximação. O sistema varia no espaço de 0 a L . Para obter estados intermediários, desenvolve-se vários modelos que descrevem a dinâmica de um ponto intermediário s_i até um ponto s_j , utilizando modelos SPC para tal.

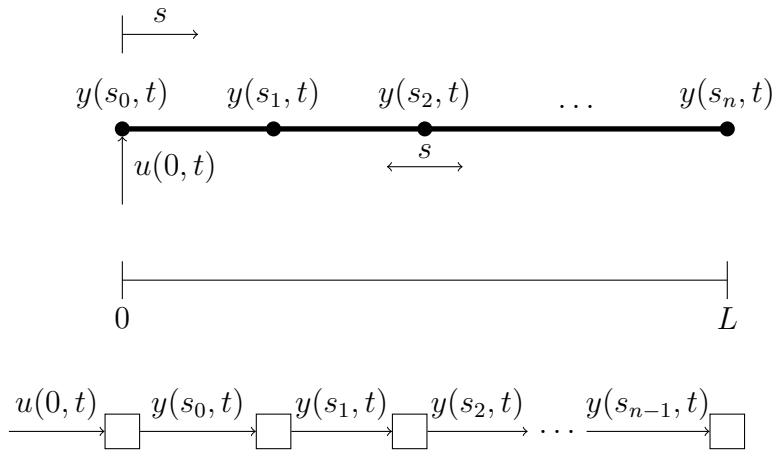


Figura 2 – Subsistemas interconectados

Já o método de modelos lineares a parâmetros variantes no espaço propõe a criação de um modelo que irá descrever a dinâmica do sistema do ponto inicial até determinado ponto espacial que é definido como uma variável, chamada de *scheduling*. Assim, alterando esta variável obtém-se como saída o estado em um ponto diferente no espaço.

A figura 3 contém o esquemático de um modelo LPV. Nele o mesmo sistema é descrito por apenas um modelo que tem um parâmetro que altera o ponto no espaço ao qual a saída

se refere. Assim pode-se obter o estado em vários pontos no espaço sem a necessidade de remodelar o sistema.

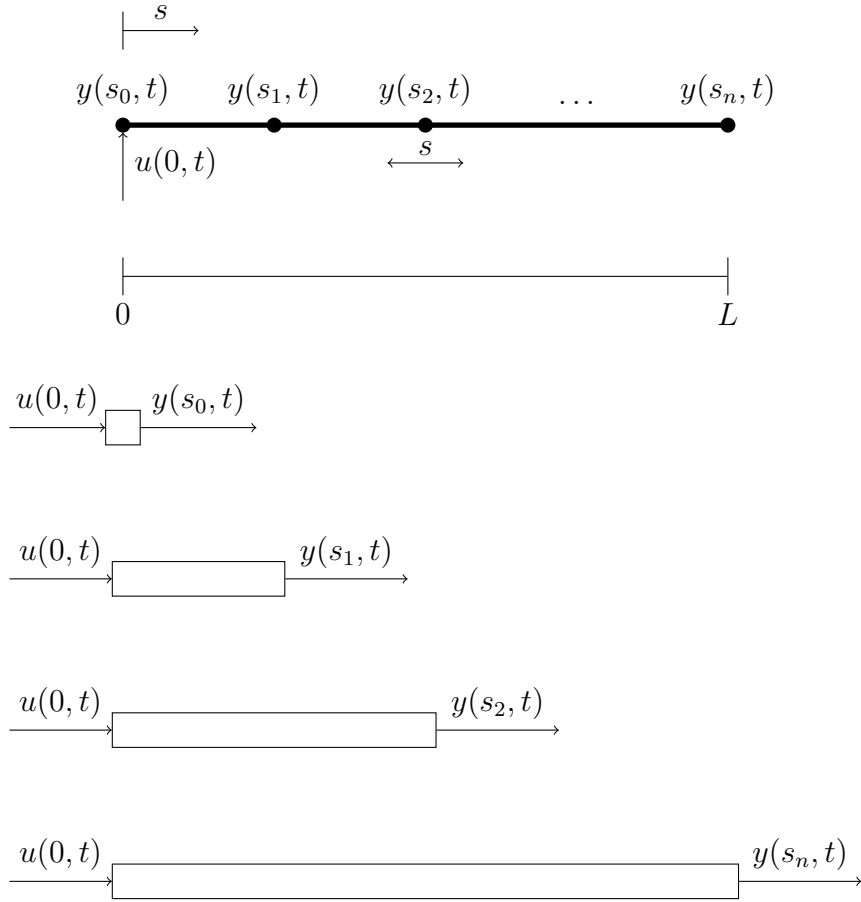


Figura 3 – Subsistema LPV

Ao combinar as duas técnicas, Barroso (2017) propõe que se use dois modelos LPV complementares como subsistemas interconectados. O primeiro modelo descreve a dinâmica do sistema do atuador até um ponto x enquanto o segundo modelo descreve a dinâmica desse ponto até uma saída fixa.

Um esquemático desta técnica é apresentado na figura 4. Nele pode-se ver que foi utilizado a técnica de subsistemas interconectados, no entanto, os subsistemas utilizados são LPV e complementares. Isso significa que ao alterar o valor da variável s_l altera-se ambos sistemas para que o primeiro forneça o estado em s_l e o segundo a saída em L , dada a saída s_l . Desta forma o ponto s_l é móvel, podendo ser alterado a qualquer momento.

A principal vantagem desta abordagem é poder utilizar um sensor na saída do segundo modelo LPV e utilizar um observador para estimar o estado de qualquer ponto entre o atuador e o sensor, selecionando o ponto através apenas da alteração da variável *scheduling*, s_l .

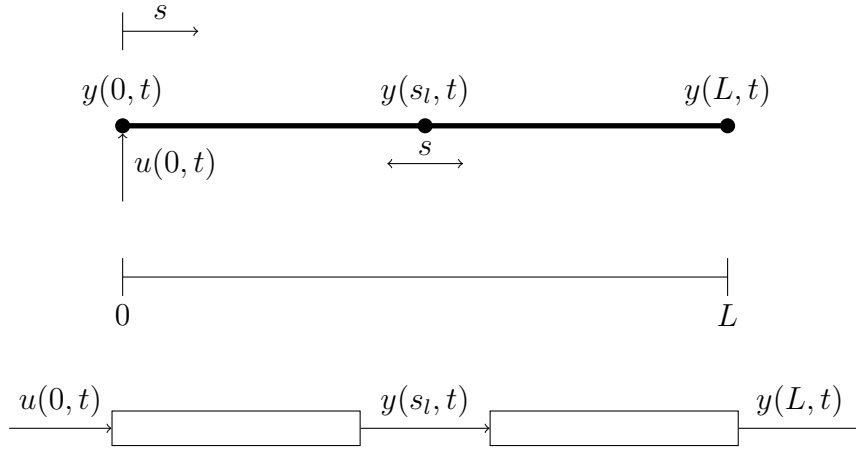


Figura 4 – Subsistemas LPV interconectados

4.2.1 Modelagem do forno por LPVs interconectados

Para modelar o sistema foram utilizados dois modelos ARMAX (Modelo autoregressivo de média variável com entradas exógenas, do inglês *Autoregressive-moving-average with exogenous inputs*), um de 0 até s_ℓ e outro de s_ℓ até L . A formulação geral para estes modelos é

$$y(s_\ell, k) = - \sum_{i_y=1}^{n_y} a_{i_y}(s_\ell) y(0, k - i_y) + \sum_{i_u=n_k}^{n_k+n_u-1} b_{i_u}(s_\ell) y(s_\ell, k - i_u) + \sum_{i_v=1}^{n_v} c_{i_v}(s_\ell) v(s_\ell, k - i_v) + v(s_\ell, k) \quad (4.27)$$

e

$$y(L, k) = - \sum_{i_y=1}^{n_y} \alpha_{i_y}(\bar{s}_\ell) y(L, k - i_y) + \sum_{i_u=n_k}^{n_k+n_u-1} \beta_{i_u}(\bar{s}_\ell) y(\bar{s}_\ell, k - i_u) + \sum_{i_v=1}^{n_v} \gamma_{i_v}(\bar{s}_\ell) v(\bar{s}_\ell, k - i_v) + v(\bar{s}_\ell, k). \quad (4.28)$$

Após um dimensionamento inicial obtém-se

$$y(s_\ell, k) = -a_1(s_\ell) y(s_\ell, k - 1) + b_1(s_\ell) y(0, k - 1) + c_1(s_\ell) v(s_\ell, k - 1) + c_2(s_\ell) v(s_\ell, k - 2) + v(s_\ell, k) \quad (4.29)$$

e

$$\begin{aligned}
 y(L, k) = & -\alpha_1(\bar{s}_\ell)y(L, k-1) + \beta_0(\bar{s}_\ell)y(\bar{s}_\ell, k) + \beta_1(\bar{s}_\ell)y(\bar{s}_\ell, k-1) \\
 & + \gamma_1(\bar{s}_\ell)v(\bar{s}_\ell, k-1) + \gamma_2(\bar{s}_\ell)v(\bar{s}_\ell, k-2) + v(\bar{s}_\ell, k),
 \end{aligned} \tag{4.30}$$

sendo os coeficientes na forma

$$a(s_\ell) = \sum_{i_a=0}^{n_a} a_{i_a} s_\ell^{i_a}. \tag{4.31}$$

Obtém-se assim as funções scheduling de 3 (4.32), 5 (4.33) e 9 (4.34) posições, obtidas utilizando a técnica de mínimos quadrados para ajustar os coeficientes.

$$\begin{aligned}
 b1 &= 1.3291 * 10^{-9} s_\ell^2 - 2.2196 * 10^{-5} s_\ell + 0.0563 \\
 a1 &= -1.5101 * 10^{-8} s_\ell^2 + 1.3591 * 10^{-5} s_\ell - 0.9118 \\
 w1 &= 1.6636 * 10^{-8} s_\ell^2 - 3.2243 * 10^{-5} s_\ell + 0.0213 \\
 \beta_0 &= -2.5839 * 10^{-4} \bar{s}_\ell + 0.9036 \\
 \beta_1 &= 2.3909 * 10^{-4} \bar{s}_\ell - 0.8628 \\
 \alpha_1 &= -1.6436 * 10^{-5} \bar{s}_\ell - 0.9559 \\
 w2 &= 4.9739 * 10^{-6} \bar{s}_\ell + 0.0029
 \end{aligned} \tag{4.32}$$

$$\begin{aligned}
 b1 &= -3.3299 * 10^{-9} * sl^2 - 1.5982 * 10^{-5} * sl + 0.05711 \\
 a1 &= -3.0164 * 10^{-8} * sl^2 + 3.3929 * 10^{-5} * sl - 0.91092 \\
 w1 &= 1.4014 * 10^{-8} * sl^2 - 2.6372 * 10^{-5} * sl + 0.01952 \\
 \beta_0 &= -4.8261 * 10^{-8} * \bar{s}_\ell^2 - 1.3554 * 10^{-4} * \bar{s}_\ell + 0.8171 \\
 \beta_1 &= 5.7969 * 10^{-8} * \bar{s}_\ell^2 + 8.1357 * 10^{-5} * \bar{s}_\ell - 0.7485 \\
 \alpha_1 &= 2.9771 * 10^{-8} * \bar{s}_\ell^2 - 8.6561 * 10^{-5} * \bar{s}_\ell - 0.9149 \\
 w2 &= 3.8259 * 10^{-9} * \bar{s}_\ell^2 - 3.1247 * 10^{-6} * \bar{s}_\ell + 0.0068
 \end{aligned} \tag{4.33}$$

$$\begin{aligned}
b1 &= -4.2490 * 10^{-12} * sl^3 + 3.4291 * 10^{-9} * sl^2 - 1.5999 * 10^{-5} * sl + 0.0568 \\
a1 &= 9.5100 * 10^{-12} * sl^3 - 4.7554 * 10^{-8} * sl^2 + 4.3890 * 10^{-5} * sl - 0.9112 \\
w1 &= -2.3370 * 10^{-11} * sl^3 + 6.6704 * 10^{-8} * sl^2 - 5.7759 * 10^{-5} * sl + 0.0237 \\
\beta_0 &= 2.0229 * 10^{-10} * \bar{s}_\ell^3 - 4.7445 * 10^{-7} * \bar{s}_\ell^2 + 1.1916 * 10^{-4} * \bar{s}_\ell + 0.7532 \\
\beta_1 &= -2.5343 * 10^{-10} * \bar{s}_\ell^3 + 6.2283 * 10^{-7} * \bar{s}_\ell^2 - 2.8674 * 10^{-4} * \bar{s}_\ell - 0.6551 \\
\alpha_1 &= -5.5237 * 10^{-11} * \bar{s}_\ell^3 + 1.4785 * 10^{-7} * \bar{s}_\ell^2 - 1.5578 * 10^{-4} * \bar{s}_\ell - 0.8992 \\
w2 &= 8.1120 * 10^{-11} * \bar{s}_\ell^3 - 1.8802 * 10^{-8} * \bar{s}_\ell^2 + 1.6462 * 10^{-4} * \bar{s}_\ell + 0.0017
\end{aligned} \tag{4.34}$$

As equações (4.30) e (4.29) podem ser reescritas no formato de espaço de estados como

$$\begin{aligned}
x(k) &= \begin{bmatrix} x_1(L, k) \\ x_2(s_\ell, k) \end{bmatrix} \\
x(k+1) &= \begin{bmatrix} -\alpha(\bar{s}_\ell) & \beta_1(\bar{s}_\ell) - \beta_0(\bar{s}_\ell)a_1(s_\ell) \\ 0 & -\alpha(s_\ell) \end{bmatrix} x(k) + \begin{bmatrix} \beta_0(\bar{s}_\ell)b_1(s_\ell) \\ b_1(s_\ell) \end{bmatrix} u(0, k) \\
y(L, k) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(k).
\end{aligned} \tag{4.35}$$

Os modelos levantados por Barroso (2017) estão discretizados com tempo de amostragem de 5 segundos.

4.3 Observadores

Observadores são modelos que reconstroem informação em tempo real. Sua função é recuperar as informações dos estados a partir do modelo do sistema, a entrada, a saída e o último estado estimado. Eles funcionam como malha fechada, utilizando o modelo para prever o próximo estado e um ganho para corrigir o erro da saída estimada em relação à saída real.

Existem vários tipos de observadores. Neste trabalho serão utilizados dois: o filtro de Kalman e o observador exponencial com fator de esquecimento. O filtro de Kalman é talvez o mais utilizado e funciona tanto como observador quanto filtro de ruídos. Porém sua formulação requer a covariância das variações dos estados. Já o observador exponencial requer apenas a covariância da saída e possui uma variável para ajustar a velocidade de convergência facilmente.

4.3.1 Filtro de Kalman

O filtro de Kalman é um observador estocástico ótimo discreto no tempo desenvolvido por Kalman (1960). Assumindo que um sistema passa a estar sujeito a ruídos gaussianos de média nula e não correlacionados v e w , podemos descrever seu comportamento como

$$\begin{aligned}x(k+1) &= Ax(k) + Bu(k) + v(k) \\y(k) &= Cx(k) + Du(k) + w(k).\end{aligned}\tag{4.36}$$

Para sistemas descritos desta forma, o Teorema 1 define o observador de Kalman.

Teorema 1 *Para qualquer sequência de entrada regularmente persistente para o par (A, C) o seguinte sistema:*

$$\begin{aligned}P(\infty) &= A(P(\infty) - P(\infty)C^T(\Gamma + CP(\infty)C^T)^{-1}CP(\infty))A^T + \Theta \\K_{ob}(\infty) &= AP(\infty)C^T(\Gamma + CP(\infty)C^T)^{-1} \\ \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + K_{ob}(\infty)(y(k) - C\hat{x}(k)) - Du(k)\end{aligned}\tag{4.37}$$

provê um observador ótimo para o sistema (4.36).

Na equação (4.37), Γ e Θ são as matrizes de covariância da saída e da variação dos estados, respectivamente. O primeiro pode ser facilmente medido. Basta realizar uma leitura fix com o sensor por várias amostras e calcular a covariância do vetor de valores resultantes. O segundo exige que seja possível medir todos os estados, o que nem sempre acontece. Sendo possível, mede-se todos os estados em um valor fixo por várias amostras e calcula-se a covariância da diferença dos estados ($x(k) - x(k-1)$ para cada estado). Caso não seja possível deve-se procurar formas de estimar a covariância ou encontrar valores manualmente que produzam resultados satisfatórios.

$P(\infty)$ e $K_{ob}(\infty)$ são os valores P e K_{ob} que estabilizam as equações, ou seja, os valores em que a saída não mais se altera. Assim o conjunto de equações (4.37) é convergente para os valores de P e K_{ob} , e esses podem ser obtidos através do cálculo iterativo de $P(\infty)$ até sua convergência.

4.3.2 Observador exponencial com fator de esquecimento

O observador exponencial com fator de esquecimento (TICLEA; BESANÇON, 2009) também é um observador do tipo Kalman. Isso pode ser visto em sua estrutura. Sua formulação é dada pelo Teorema 2.

Teorema 2 *Dado os sistema (4.36), e assumindo que uma sequência de entrada, $u(k)$, é regularmente persistente para o par (A, C) e torna A invertível, então, um observador globalmente exponencialmente convergente é dado por:*

$$\begin{aligned} P(\infty) &= \delta^{-1} A(P(\infty) - KCP(\infty))A^T \\ K_{ob}(\infty) &= AP(\infty)C^T(\Gamma + CP(\infty)C^T)^{-1} \\ \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + K_{ob}(\infty)(y(k) - C\hat{x}(k)) - Du(k). \end{aligned} \quad (4.38)$$

Na equação (4.38), δ é um número real que controla a velocidade de convergência do observador. Quanto menor δ , mais rápido o observador converge.

4.3.3 Observabilidade

Observabilidade é a medida de quão bem os estados internos de um sistema podem ser estimados a partir de suas saídas. Formalmente um sistema é dito observável se para qualquer sequência de estados e sinais de controle, o estado atual pode ser determinado em tempo finito usando apenas as saídas. Se o sistema não é observável, então existe pelo menos um estado que não pode ser determinado apenas pela saída.

Para sistemas invariantes no tempo em espaço de estados com n estados, se o posto da matriz L_o da equação (4.39) for igual a n , o sistema é dito observável. A ideia por trás deste teste é que, se a matriz contém n linhas linearmente independentes, então os n estados podem ser vistos como combinações lineares da saída.

$$L_o = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (4.39)$$

Resultados e Discussões

Neste capítulo são apresentados os resultados parciais obtidos, separados por tópicos conforme o cronograma apresentado na proposta.

5.1 Estudos teóricos

5.1.1 Controle preditivo por modelo

O livro-texto utilizado para os estudos do controle preditivo por modelo foi o do Wang (2009), com as outras referências sendo usadas de forma complementar. Para a execução deste projeto propõe-se o estudo dos capítulos 1 a 4, que tratam do caso discreto.

Foram estudados os capítulos 1 e 2, que apresentam o conceito do controlador e a formulação com restrições, respectivamente. Como os estudos teóricos de MPC foram iniciados antes da alteração da eletrônica do forno, a técnica foi testada utilizando os tanques comunicantes (Figura 5) presentes no Laboratório de Sinais e Sistemas.

Os tanques utilizados são o 1º e 2º, que estão localizados na parte superior e marcados como T1 e T2. A bomba insere água no primeiro tanque e esta escoar para o segundo através de uma válvula localizada na parte inferior. A água é então removida através de uma válvula no fundo do tanque 2. O objetivo de controle é o nível do tanque 2.

Uma simulação do controlador MPC controlando o modelo desse sistema pode ser visto na Figura 6. Observa-se que o sinal de controle é mantido em 100% por quase todo o transitório, e é então abaixado para o valor de equilíbrio, fazendo com que o tanque encha de forma rápida mas minimizando o overshoot.

Ao inserir este controlador no sistema real (Figura 7) obtém-se uma resposta parecida



Figura 5 – Sistema de tanques comunicantes

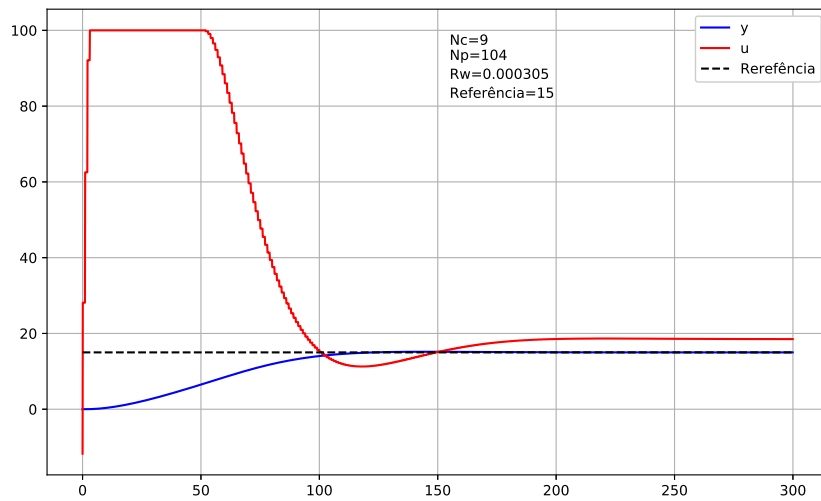


Figura 6 – Simulação do modelo dos tanques com controlador MPC

com a da simulação, porém mais lenta. Essa diferença pode ser explicada por erro de modelagem e pelo fato que o sistema real não é linear, sendo o modelo linearizado. No entanto a resposta continua satisfatória.

Como foi utilizado um observador, também pode-se perceber os efeitos do erro de modelagem. Os estado estimado, x_1 , embora apresente a mesma dinâmica, não apresenta o mesmo ganho que o estado real, h_1 . É interessante notar, no entanto, que esta diferença não atrapalha o controlador, que continua capaz de seguir a referência.

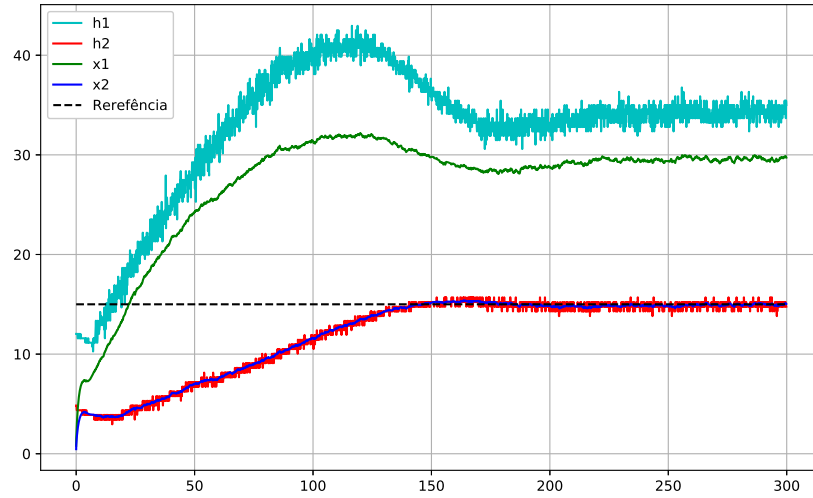


Figura 7 – MPC aplicado ao sistema real

5.1.2 Modelagem térmica e SPD

Os resultados obtidos por Barroso (2017) foram simulados e utilizou-se o modelo SPD apresentado com os observadores de Kalman e exponencial com fator de esquecimento. Foi possível ver nos estados a saída variando conforme a distância do atuador aumenta, como pode ser visto na Figura 8. Os observadores também estimam com precisão os estados do sistema simulado, conforme Figura 9.

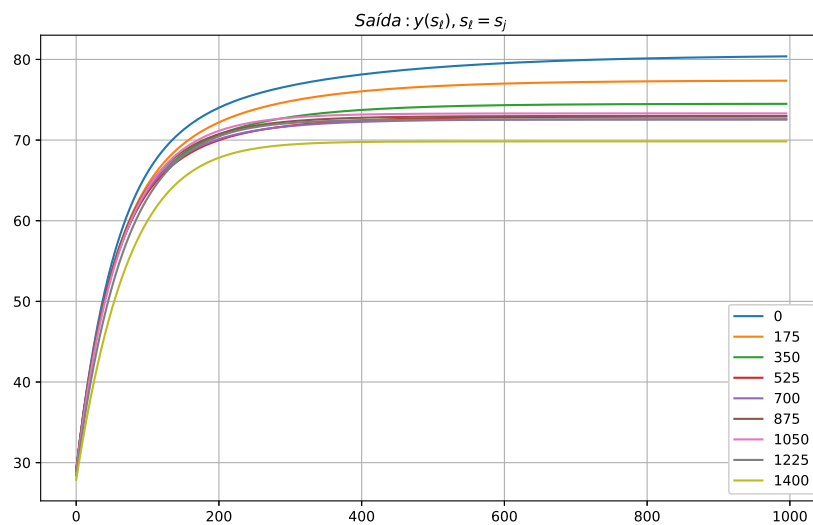


Figura 8 – SPD simulado

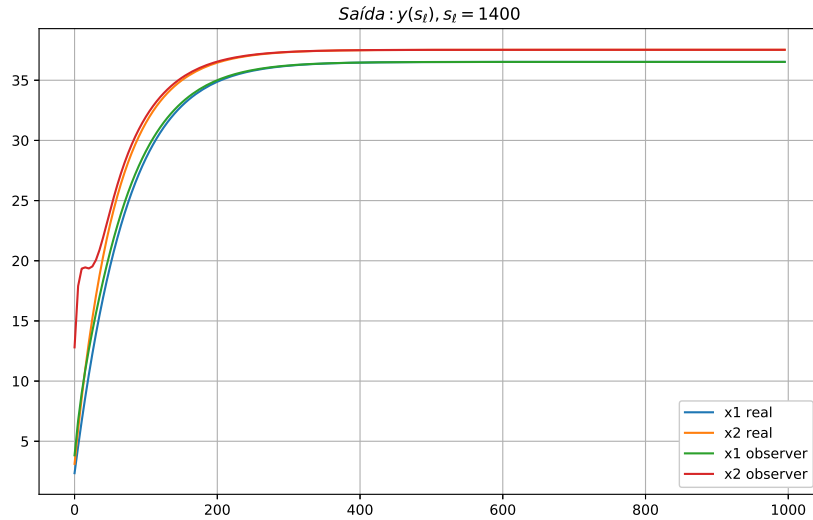


Figura 9 – SPD simulado com observador exponencial

5.2 Modificação do hardware e implantação da plataforma

Ao estudar o circuito de acionamento e sensoriamento percebeu-se que não há a necessidade de alterar o circuito. Devido a forma como o mesmo está montado, foi necessário apenas a reprogramação dos microcontroladores presentes para possibilitar a comunicação direta através de cabos USB. Após a reprogramação foi possível acionar o circuito de potência e ler os valores dos sensores.

A instalação do Raspberry não foi feita pois o mesmo não foi adquirido a tempo. No entanto, as modificações necessárias para a sua inserção já foram executadas em outros projetos. Tais modificações compreendem principalmente mudanças no software *moirai* e já foram publicadas.

5.3 Calibrações, testes e certificações

O funcionamento do circuito eletrônico foi feito primeiramente acionando o sistema desenvolvido por Barroso (2017). Após verificar que este era capaz de acionar as resistências e ler valores de todos os sensores, foi feita a modificação dos códigos dos microcontroladores para permitir a comunicação com a plataforma e testou-se novamente com o novo código. Verificou-se a necessidade de recalibrar os sensores e reordená-los, identificando

qual estava conectado a qual porta. Devido a forma como se dá o acionamento não é necessária a calibração dos atuadores.

Para a calibração dos sensores foi utilizada uma garrafa térmica perfurada e uma lâmpada incandescente. A mesma foi ligada e foram feitas medições das temperaturas dos sensores bem como de um termômetro industrial a cada 10 segundos. Foram coletados 160 pontos. Utilizou-se então a técnica do mínimos quadrados para encontrar uma expressão de correção linear, $ax + b$, cujos coeficientes podem ser vistos na Tabela 3.

Tabela 3 – Coeficientes da calibração

| | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | S10 |
|---|-------|------|------|------|------|------|------|------|------|------|
| a | 0.85 | 0.82 | 0.82 | 0.78 | 0.79 | 0.83 | 0.76 | 0.74 | 0.81 | 0.79 |
| b | -0.65 | 3.32 | 3.40 | 3.06 | 4.96 | 2.12 | 5.24 | 6.21 | 1.39 | 4.55 |

A validação dos modelos e observadores ainda está sendo executada, pois foram encontradas dificuldades durante a validação. O problema encontrado é que o observador não está estimando o segundo estado corretamente. O proponente buscará resolver este problema juntamente com seu orientador e o author do modelo.

5.4 Melhorias na plataforma

Foram feitas várias modificações na plataforma. A lista completa pode ser vista nos *commits* de cada projeto, publicados desde 1º de março de 2018:

- Lachesis: <https://github.com/acristoffers/Lachesis/commits/master>
- moirai: <https://github.com/acristoffers/moirai/commits/master>
- ahio: <https://github.com/acristoffers/ahio/commits/master>

As modificações que se destacam são:

- implementação de um mecanismo de *backup* de todo o banco: o uso de múltiplos computadores na mesma planta pode ser facilitado com a opção de realizar backup de todo o banco. Isso também permite a liberação de espaço no sistema sem a perda de dados, já que pode-se guardar os dados antigos e apagar os testes, deixando a interface mais limpa;

- implementação de importação e exportação de configuração de hardware: também interessante quando várias pessoas utilizam o mesmo hardware. Mudanças, como novas calibrações, podem ser compartilhadas de forma simples e novos usuários podem começar a utilizar a planta rapidamente, apenas importando uma configuração já testada;
- seleção das variáveis a serem exibidas em gráficos: durante o desenvolvimento inicial não se pensou na possibilidade do usuário precisar salvar dezenas ou centenas de variáveis. Toda variável salva geraria um gráfico para o usuário, sempre. Durante os testes com casos reais, realizados por alunos de graduação e mestrado, observou-se a necessidade de limitar o número de gráficos gerados sem limitar o número de variáveis salvas. Isso foi feito colocando uma opção de selecionar quais gráficos serão exibidos;
- opção de clonar um teste ou controlador: durante os testes também percebeu-se a necessidade de duplicar um teste inteiro para modificar apenas um valor de variável, ou outra modificação pequena;
- melhoria da forma de apresentação de erros no terminal: os erros apresentados no terminal não eram explicativos, não mostrando, por exemplo, onde ocorreu o erro. Agora os erros trazem o escopo e a linha do erro, bem como a mensagem principal (erro mais relevante). Esse erro, na versão 1.0.11 do Lachesis, também é exibido para o usuário na interface, no componente Gráficos;
- melhoria de velocidade na recuperação de dados do banco: foram utilizadas funções disponíveis em versões mais recentes do MongoDB para melhorar a filtragem e transformação dos dados retornados. Esses eram feitos anteriormente em Python, com uma implementação mais lenta que aquela do banco de dados;
- inserção de um novo banco de dados: MySQL (necessário no Raspberry), sua necessidade veio da modificação anterior. Ao atualizar o banco de dados percebeu-se que as versões mais novas não mais suportam sistemas operacionais de 32bits, por isso foi necessário encontrar um banco de dados capaz de executar em um Raspberry Pi de 32bits. A seleção do MySQL se deu pela sua disponibilidade e atualização no Raspbian, sistema operacional do Raspberry e por sua implementação em lin-

guagem C/C++. Outros bancos de dados não estruturados do tipo *document store* foram considerados, mas a falta de documentação, api pobre e/ou implementação em linguagem interpretada (mesmo que por máquina virtual) os fizeram menos interessantes;

- implementação de busca por drivers (ahio) em caminhos arbitrários: a biblioteca ahio implementa alguns drivers padrão que estão presentes em seu código fonte. No entanto, faz-se necessário a implementação de drivers específicos, que não devem ser inseridos e distribuídos na biblioteca por não serem úteis em outros contextos, por exemplo, o driver do forno e os drivers utilizados pelos alunos Affonso Salomão e Bernardo Amin em seus TCCs. Para solucionar esse problema criou-se a possibilidade de procurar por drivers em caminhos informados pelo usuário, exportando-se funções para adicionar e remover caminhos do *PATH*. No aplicativo *moirai* foi criada a opção de se definir uma variável de ambiente denominada *AHIO_PATH* contendo caminhos separados por “:” (dois pontos), conforme padrão *POSIX*.

Conclusão

Apesar do contratempo encontrado durante a validação dos modelos e observadores, o progresso dos trabalhos pode ser considerado satisfatório. Foi possível testar o controlador em outro sistema e perceber suas vantagens, além de experimentar duas formulações diferentes de observadores. As modificações feitas na plataforma já foram utilizadas por membros do Grupo de Modelagem e Controle de Sistemas Mecatrônicos que reportaram benefícios advindos das melhorias.

No segundo semestre será estudada a formulação por funções de Laguerre do controlador MPC e projetados controladores PID e MPC para o forno. A implementação de melhorias na plataforma será continuada.

Bibliografia

BARROSO, N F. **Estratégia de monitoramento de sistemas distribuídos baseada em observadores do tipo Kalman**. 2017. Master Thesis – CEFET-MG. Disponível em: <<https://goo.gl/oDYCdB>>.

_____. **Instrumentação Virtual Aplicada à Automação de um Sistema Térmico para Experimentação Via Web**. [S.l.: s.n.], 2015. Disponível em: <<https://goo.gl/uff757>>.

BLACK, H. S. Stabilized Feedback Amplifiers. **Bell System Technical Journal**, v. 13, n. 1, p. 1–18, 1934. ISSN 15387305. DOI: 10.1002/j.1538-7305.1934.tb00652.x.

BODE, Hendrik Wade. Feedback Amplifier Design. **Bell System Technical Journal**, v. 19, n. 1, p. 42, 1940.

BRYSON, Arthur E.; HO, Yu-Chi; SIOURIS, George M. **Applied Optimal Control: Optimization, Estimation, and Control**. Revised. [S.l.]: Taylor & Francis, 1979. v. 9, p. 366–367. ISBN 9780891162285. DOI: 10.1109/TSMC.1979.4310229. Disponível em: <<http://ieeexplore.ieee.org/document/4310229/>>.

CAIRANO, S Di. An industry perspective on MPC in large volumes applications: Potential Benefits and Open Challenges. In: PROC. 4th IFAC Nonlinear Model Predictive Control ... [S.l.: s.n.], 2012. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.374.9189%7B%5C%7Drep=rep1%7B%5C%7Dtype=pdf>>.

CLARKE, D W. PID algorithms and their computer implementation. **Transactions of the Institute of Measurement and Control**, v. 6, n. 6, p. 305–316, 1984. ISSN

0142-3312. DOI: 10.1177/014233128400600605. Disponível em:

<<http://tim.sagepub.com/content/6/6/305.abstract>>.

CUTLER, C.R.; RAMAKER, B.L. Dynamic matrix control- A computer control algorithm. **Joint Automatic Control Conference**, v. 1, 1980.

DORF, Richard C.; BISHOP, Robert H. **Modern Control Systems**. [S.l.]: Pearson, 2010. p. 1104. ISBN 9780136024583. Disponível em: <<https://goo.gl/kKU1vj>>.

KALMAN, R. E. A New Approach to Linear Filtering and Prediction Problems. **Journal of Basic Engineering**, American Society of Mechanical Engineers, v. 82, n. 1, p. 35, mar. 1960. ISSN 00219223. DOI: 10.1115/1.3662552. Disponível em: <<http://fluidsengineering.asmedigitalcollection.asme.org/article.aspx?articleid=1430402>>.

LIBERTY, S. **Modern control engineering**. [S.l.]: Pearson, 1972. v. 17, p. 419–419. ISBN 0136156738. DOI: 10.1109/TAC.1972.1100013. arXiv: 0605511 [cond-mat]. Disponível em: <<http://ieeexplore.ieee.org/document/1100013/>>.

LYAPUNOV, Aleksandr Mikhailovich. Problème Général de la Stabilité du Mouvement. **Annales de la Faculté des sciences de Toulouse : Mathématiques**, GAUTHIER-VILLARS, IMPRIMEUR-EDITEUR ; ED. PRIVAT, IMPRIMEUR-LIBRAIRE, v. 9, p. 203–474, 1892. Disponível em: <<http://eudml.org/doc/72801>>.

MARTINS, Fg. Tuning PID controllers using the ITAE criterion. **International Journal of Engineering Education**, v. 21, n. 3, p. 867–873, 2005. Disponível em: <<http://pessoal.utfpr.edu.br/avargas/Ijee1673.pdf>>.

MORARI, Manfred; LEE, Jay H. Model predictive control: past, present and future. **Computers I& Chemical Engineering**, v. 23, n. 4–5, p. 667–682, 1999. ISSN 0098-1354. DOI: 10.1016/S0098-1354(98)00301-9. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0098135498003019>>.

PATWARDHAN, Sachin C. **A Gentle Introduction to Model Predictive Control (MPC) Formulations based on Discrete Linear State Space Models**. [S.l.: s.n.], 2014. Disponível em: <https://www.iitk.ac.in/tkic/workshop/control%20techniques/ppt/patvardhan/LQG%7B%5C_%7DMPC%7B%5C_%7DNotes%7B%5C_%7DIITK%7B%5C_%7D2016.pdf>.

SILVA, J V Valle. **Modelagem Matemática a parâmetros distribuídos e Controle em Malha Fechada de um sistema de aquecimento de ar**. [S.l.: s.n.], 2014. Disponível em: <<https://goo.gl/LTFRw8>>.

ȚICLEA, Alexandru; BESANÇON, Gildas. State and parameter estimation via discrete-time exponential forgetting factor observer. **IFAC Proceedings Volumes**, Elsevier, v. 42, n. 10, p. 1370–1374, jan. 2009. ISSN 1474-6670. DOI: 10.3182/20090706-3-FR-2004.00228. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1474667016388425>>.

WANG, Liuping. **Model Predictive Control System Design and Implementation Using MATLAB**. 1. ed. [S.l.]: Springer-Verlag London, 2009. p. 403. (Advances in Industrial Control). ISBN 9781848823303. DOI: 10.1007/978-1-84882-331-0. arXiv: arXiv:1011.1669v3. Disponível em: <<https://goo.gl/czFKuq>>.

ZHANG, Xi. **Fast MPC Solvers for Systems with Hard Real-Time Constraints**. [S.l.: s.n.], 2016. p. 86.