INTRODUÇÃO

CONTEXTUALIZAÇÃO

[Tentar usar introdução das versões antigas]

PROBLEMA E JUSTIFICATIVA:

Frente à importância de sistemas supervisórios no âmbito da Automação e Controle de processos, e dos altos custos de licenças de alguns softwares como MatLab e SIMATIC WinCC, surgiu a ideia da construção de uma plataforma que utilize uma linguagem gratuita e *open-source* e sirva de alternativa a aplicações deste ínterim. A mesma seria utilizada para comunicar-se com qualquer controlador conectado por porta serial ao computador que a executasse, registrando em forma de gráficos e valores estatísticos variáveis inerentes a sistemas mecânicos, elétricos ou quaisquer outros monitorados por sensores conectados a tal controlador, ou modelados por funções de transferências. Além disto, teria código livre e aberto, tanto para o estudo e aprendizado dos estudantes da universidade, como para futuras melhorias e inclusão de novas funcionalidades.

OBJETIVOS

OBJETIVOS GERAIS

Desenvolver em linguagem Python uma GUI (Graphical User Interface) capaz de se comunicar com dispositivos externos que lhe fornecessem dados numéricos para plotagem em sua interface, de forma que seja possível a análise de respostas de sistemas físicos diversos e comparação com modelagens criadas pelos usuários deste sistema.

OBJETIVOS ESPECÍFICOS

Com o foco nos objetivos supracitados, foram levantados os seguintes pequenos *milestones* a serem alcançados no decorrer do desenvolvimento do sistema:

* Realizar pesquisa bibliográfica das bibliotecas Python utilizadas
* Desenhar a(s) tela(s) do sistema
* Programar os eventos que coordenam o funcionamento do sistema
* Configurar a comunicação com dispositivos externos (preferencialmente tomando o controlador Arduino como base)
* Confeccionar um tutorial objetivo relativo ao funcionamento do sistema

ESTRUTURA DO TRABALHO

Este trabalho foram divididos em

* No capítulo 1 o tema principal foi contextualizado de acordo com a tecnologia atual, o problema a ser solucionado foi apresentado, e os objetivos foram esclarecidos e segmentados em um passo-a-passo.
* No capítulo 2 consta uma revisão bibliográfica relativa aos assuntos que cercam o tema principal deste trabalho, a iniciar por uma descrição curta dos softwares SCADA existentes, das tecnologias que podem ser utilizadas para a criação de uma GUI e finalmente um resumo da linguagem selecionada para a construção da interface
* O capítulo 3 trata da programação da plataforma em si, em linguagem Python, mostrando sua arquitetura e convenções empregadas com figuras e esquemas.
* O capítulo 4 apresenta o sistema já pronto e aborda as lições aprendidas no decorrer de seu desenvolvimento, além de outras discussões relativas à sua programação.
* No capítulo 5 este trabalho se encerra em forma de um texto conclusivo, onde, entre outros assuntos, são abordadas possíveis futuras melhorias ao sistema criado.

REVISÃO BIBLIOGRÁFICA

MODELAGEM DE PROCESSOS

A dinâmica de muitos sistemas mecânicos, elétricos, térmicos, econômicos, biológicos ou outros pode ser descrita em termos de equações diferenciais. Essas equações diferenciais são obtidas pelas leis físicas que regem dado sistema - por exemplo, as leis de Newton para sistemas mecânicos e as leis de Kirchhoff para sistemas elétricos. Um modelo matemático não é o único para determinado sistema. Um sistema pode ser representado de muitas maneiras diferentes e. portanto, pode ter vários modelos matemáticos, dependendo da perspectiva a ser considerada. (OGATA KATSUHIRO, 2010)

Uma das formas mais comuns de representação de um sistema genérico é por funções de transferências. Sendo *y(t)* a função que descreve uma saída de um processo, e *x(t)* a de uma entrada deste processo, a função de transferência *G(s)* traduz a influência de *x* em *y*. Assim, a função *G(s)* advém das equações físicas descritivas de um processo e, de maneira geral se torna mais complexa quanto mais equações e parâmetros são considerados. Como um exemplo, ao modelar a queda de um corpo livre, tomando como variável de saída sua posição, uma função de transferência simples consideraria apenas a influência da gravidade sobre o corpo, e a mesma pode se tornar mais complexa se considerasse o atrito e resistência com o ar.

A modelagem de um processo pode traduzi-lo em um sistema linear ou não linear.

Um sistema é dito linear se o princípio da superposição se aplicar a ele. Este princípio afirma que a resposta produzida pela aplicação simultânea de duas funções de determinação diversas é a soma das duas respostas individuais. Então, para o sistema linear, a resposta a diversas entradas pode ser calculada tratando uma entrada de cada vez e somando os resultados. Esse é o princípio que permite construir soluções complicadas para equações diferenciais lineares a partir de soluções simples. (OGATA KATSUHIRO, 2010)

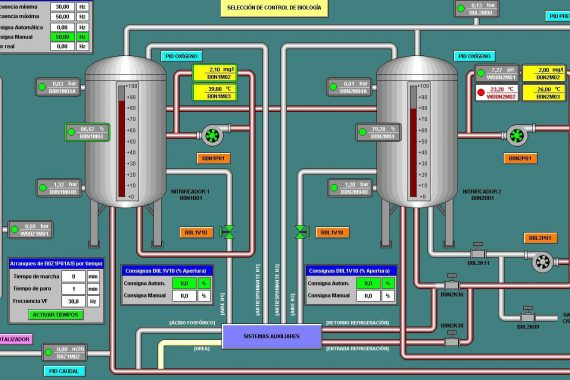
Sistemas não lineares são, em geral, mais difíceis de modelar e de controlar. Assim, podem ser empregadas algumas técnicas para transformá-los em sistemas lineares. Uma delas é a chamada linearização, que emprega a série de Taylor, truncando a série no segundo termo e obtendo assim, uma função linearizada em torno de um determinado ponto de operação do processo. Quanto mais as variáveis do sistema linearizado se afastarem deste ponto de operação, maior será o erro deste sistema, em relação ao sistema não linear. Outra técnica é partir o sistema não linear com uma entrada qualquer, e pela análise do gráfico de resposta projetar um sistema linear que seja o mais similar possível ao primeiro.

Talvez o maior benefício da modelagem de processos para o setor industrial seja a possibilidade de projetar sistemas de controle mais eficientes, eliminando a necessidade de gastar com testes em campo. Softwares como MATLAB e GNU Octave permitem que, a partir de funções (no tempo ou de transferência) um processo seja modelado e seu comportamento, dada entradas também configuradas, seja simulado.

SCADA (Supervisory Control And Data Acquisition)

Os sistemas supervisórios podem ser considerados como o nível mais alto de IHM, pois mostram o que está acontecendo no processo e permitem ainda que se atue neste. A evolução dos equipamentos industriais, com a introdução crescente de sistemas de automação industrial, tornou complexa a tarefa de monitorar, controlar e gerenciar esses sistemas. (MACHADO MARTINS, 2012)

Sistemas SCADAs são responsáveis por buscar informações de controladores e equipamentos diversos de automação e manipular estas informações de diversas maneiras. As aplicações mais simples se constituem na visualização dinâmica destes dados através de objetos, mostradores, cores, entre outros meios dispostos em telas pré programadas. Uma estratégia, por exemplo, é representar todo um sistema por imagens já embutidas na biblioteca do software editor, e incluir o máximo de informações importantes referentes ao mesmo em uma única tela detalhada. Outra seria agrupar as informações por temática, e mostrar diversas telas menos detalhadas, que alternem entre si de acordo com um temporizador interno.

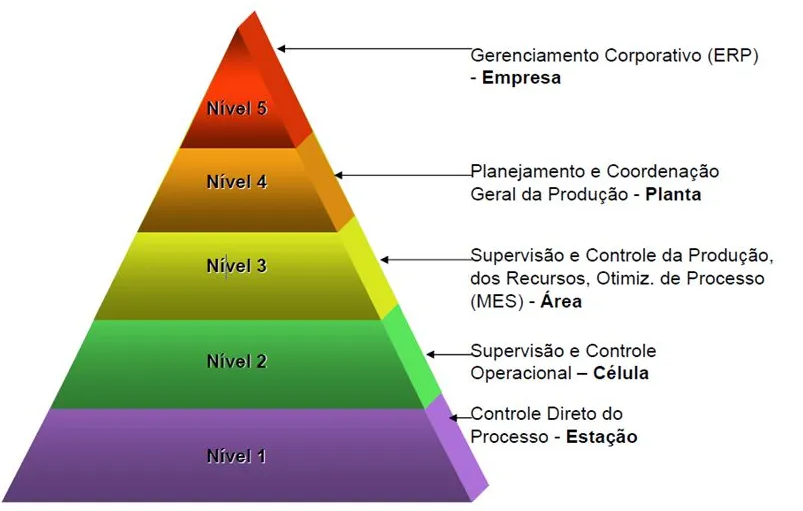


Fonte: <https://www.agaads.com/service/scada-system/>

Os dados adquiridos podem ser manipulados de modo a gerar valores para parâmetros de controle como “set-points”. Os dados são armazenados em arquivos de dados padronizados, ou apenas utilizados para realização de uma tarefa. Esses dados que foram armazenados em arquivos poderão ser acessados por programas de usuários para realização de cálculos, alteração de parâmetros e de seus próprios valores. (MACHADO MARTINS, 2012).

Outro emprego comum de sistemas SCADA, é de serem responsáveis por informar valores de setpoint aos controladores acoplados a um processo, os quais podem advir de um cálculo computacional ou manualmente, por um operador. É possível ainda o sistema ser responsável pelo controle, enviando aos atuadores conectados somente o sinal de controle. Esta estratégia, no entanto, não é recomendada, por questões de confiabilidade da transmissão de dados e velocidade de processamento.

Por ser executado geralmente em um computador comum, a palavra chave de um sistema supervisório é flexibilidade. Um sistema SCADA deve ser capaz de se comunicar por diversos protocolos com diversos dispositivos, e adicionalmente disponibilizar os valores lidos para outros usuários, não somente os que têm acesso às telas. Por possuir esta funcionalidade, software SCADAs ultrapassam o nível 2 na pirâmide de automação, referente à supervisão e controle de uma célula específica de produção, chegando ao nível seguinte, de supervisão da produção, pois agrupa as informações de diversos controladores e sensores em um só local, gerando suporte para ações de gerência.



Fonte: <https://www.logiquesistemas.com.br/blog/piramide-de-automacao-industrial/attachment/354/>

A fim de atingir uma maior compatibilidade com programas e usuários externos, a maior parte dos softwares supervisórios permitem de forma simples a criação de um banco de dados para os valores monitorados. Os mesmos podem ser exportados em forma de relatórios em layouts já embutidos e utilizados como insumos para tomada de decisões e cálculo de performance.

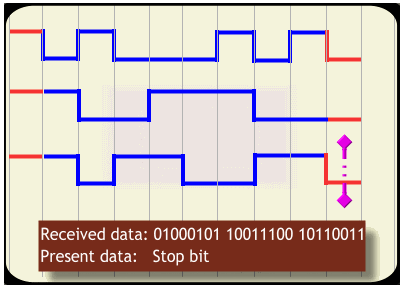
Segundo Leandro, et al, dentre os principais benefícios do uso de sistemas de supervisão podem-se citar: informações instantâneas, redução no tempo de produção, redução no custo de produção, precisão das informações, detecção de falhas, aumento da qualidade e aumento da produtividade.

COMUNICAÇÃO SERIAL

Comunicação serial é um meio simples de dois equipamentos trocarem informação em formato de uma série de bits. Ela ocorre através de um único cabo ou pino em um circuito integrado, sendo este um dos motivos da sua popularidade: o baixo custo necessário para a montagem da infraestrutura. A desvantagem disto é a perda em velocidade, pois transmite-se apenas um bit de cada vez, enquanto que a chamada comunicação paralela, que utiliza mais canais de comunicação e pode transmitir vários bits simultaneamente.

Diversos equipamentos eletrônicos utilizam a comunicação serial, como mouses e teclados. O conhecido controlador Arduino UNO também permite a troca de bits por meio de uma porta serial. Alguns protocolos de comunicação modernos utilizam a comunicação serial, como Ethernet e Profibus.

Um modelo de transmissão para um *byte* (8 bits) de informação por porta serial se constitui em uma onda digital cujos formato se traduz em um bits 1 ou 0. O primeiro bit marca o início da transmissão, seguido por 8 bits relativos à informação enviada, um bit opcional de paridade (que torna o dados mais confiáveis), e um último bit que encerra o bloco de informação. Protocolos de comunicação diferentes podem acrescentar ou remover características à sequência transmitida, seja para assegurar a integridade dos dados ou para aumentar a velocidade de comunicação.



Exemplo de transmissão de uma sequência de 3 bytes

Fonte: <http://www.c2o.pro.br/automacao/ar01s11.html>

Por comunicar dois equipamentos distintos, com diferentes arquiteturas, se fazem necessárias medidas que contornam a diferença entre os *clocks* de ambos, em outras palavras, a diferença entre a velocidade de transmissão e a de leitura dos bits recebidos. Uma delas é a utilização de um *buffer*, um espaço de memória na porta receptora, que guarde rapidamente os bits transmitidos, para posterior leitura e tratamento dos mesmos por parte do processador da máquina. Quando este buffer está próximo de encher, o receptor pode fechar o barramento serial via hardware ou software, para impedir a perda de informação durante sua transmissão.

PYTHON

Com traços da linguagem C e Perl, Python foi criada por Guido van Rossum no início da década de 90. É uma linguagem de alto nível, orientada a objetos e com tipagem dinâmica e forte. As definições de escopo e blocos de código são representadas por indentações, o que torna o código mais organizado e visualmente aprazível. Além disto permite interoperabilidade com outras linguagens. Por exemplo, utilizando a ferramenta Cython é possível, a partir de um código Python, gerar um código equivalente em C. Existem funções, inclusive, que são desenvolvidas em C, a fim de agilizar o processamento de grandes bases de dados, mas implementadas em Python.

No âmbito acadêmico, Python apresenta boas vantagens. Não só é considerada simples fácil de aprender, como é gratuita e open source. Logo, seus usuários e clientes não têm custos com licenças e seus desenvolvedores podem usar livremente códigos publicados por terceiros, que geralmente se apresentam de fácil acesso na internet, e adaptá-los às suas necessidades. Em uma enquete realizada pelo conhecido fórum da comunidade de computação *Stack Overflow*, Python foi considerada a 3ª “linguagem mais amada” pelo público.

Pela facilidade de compartilhamento e comunidade crescente, existem diversas bibliotecas de Python que podem ser baixadas com o simples comando no terminal *pip install*, configurado na instalação da linguagem. Como alguns exemplo, cita-se as *libs* TensorFlow, para criação de modelos de dados, SQLAlchemy que permite criação e acesso a bancos de dados leves e NumPy, uma poderosa ferramenta para cálculos matriciais.

.

REFERÊNCIAS BIBLIOGRÁFICAS

MACHADO MARTINS, G. Princípios de automação industrial. Santa Maria: UFSM. 2012

ROGGIA LEANDRO; FUENTES RODRIGO, C. Automação industrial. Santa Maria UFSM. 2016 disponível em <<https://www.ufsm.br/unidades-universitarias/ctism/cte/wp-content/uploads/sites/413/2018/12/06_automacao_industrial.pdf>>

OGATA KATSUHIRO, Engenharia de Controle Moderno, 5ª Edição, Editora Pearson Education, 2010